

**Міністерство освіти і науки України
Тернопільський національний технічний
університет імені Івана Пулюя**

**Кафедра автоматизації технологічних
процесів і виробництв**

Проектування мікропроцесорних систем керування

Навчальний посібник

Тернопіль

2015

УДК 681.325
ББК 32.973.2
М42

*Рекомендовано до друку вченою радою Тернопільського національного технічного
університету імені Івана Пулюя
протокол № 15 від 18 грудня 2014 р.*

Рецензенти:

*О.М. Березький, докт. тех. наук, проф.,
А.М. Лупенко, докт. тех. наук, доцент.*

М42 Медвідь В.Р. Проектування мікропроцесорних систем керування : навчальний посібник, перевидання / Медвідь В.Р., Пісціо В.П., Козбур І.Р. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 360 с.

ISBN 978-966-305-064-5
ISBN 978-966-305-064-5

Одним із основних напрямків автоматизації сучасних технологічних процесів є розроблення та впровадження мікропроцесорних систем керування, що дозволяє ефективно вирішувати завдання, поставлені перед автоматизацією в різних галузях народного господарства.

Зважаючи на багатоцільовий характер розроблення та виробництва засобів автоматизації технологічних процесів, найважливіше значення набуває проблема подальшого підвищення їх ефективності, яка визначається системою структурно-функціональних характеристик мікропроцесорних систем керування.

За останні роки в мікроелектроніці бурхливого розвитку набув напрямок, пов'язаний з випуском мікроконтролерів, призначених для «інтелектуалізації» різного устаткування. Мікроконтролери є приладами, які конструктивно виконані у вигляді інтегральних схем і містять у собі всі основні частини типової мікроЕОМ: мікропроцесор, пам'ять програм і пам'ять даних, а також програмовані інтерфейсні схеми для зв'язку із зовнішнім середовищем.

Використання мікроконтролерів у системах управління забезпечує досягнення винятково високих показників ефективності при досить низькій вартості (у багатьох застосуваннях система може складатися тільки з одного мікроконтролера). Тому можна констатувати, що мікроконтролери, мабуть, не мають нормальної альтернативної елементної бази для побудови керуючих або регулюючих систем.

На сьогодні більше двох третин світового ринку мікропро-цесорних коштів складають саме мікроконтролери.

Структурна організація, набір команд і апаратно-програмні засоби вводу-виводу інформації в мікроконтролерах найкраще пристосовані для вирішення завдань управління й регулювання в приладах та пристроях і системах автоматики, а не для вирішення завдань опрацювання даних.

Саме їх використання в розробленні систем керування дозволяє створювати недорогі вискоєфективні мікропроцесорні пристрої та забезпечувати їх упровадження не тільки у виробничій сфері, а й побуті.

Даний посібник допоможе вивченню цих питань і є необхідним для підготовки досвідчених фахівців у галузі проектування систем керування. В ньому розглянуто особливості систем різних рівнів складності та призначення, принципи архітектурних рішень, способи і засоби організації обміну інформацією.

Особливу увагу приділено принципам організації складних і гнучких мікропроцесорних систем, що дозволяють вирішувати найскладніші завдання.

Ще одне завдання посібника – навчання навичкам проектування систем на основі мікропроцесорних комплектів та мікроконтролерів як найбільш поширеного типу мікропроцесорних систем. Для її вирішення наведено опис мікроконтролерів сімейства PIC, а також спеціальних програмних засобів проектування, розглянуто приклади розв'язання задач проектування кількох пристроїв.

Навчальний посібник буде корисний студентам, інженерно-технічним працівникам, науковцям та аспірантам відповідних напрямків.

УДК 681.325
ББК 32.973.2

© В.Р. Медвідь, В.П. Пісціо, І.Р. Козбур 2015

© Тернопільський національний технічний

ISBN 978-966-305-064-5

університет імені Івана Пулюя, 2015

ISBN 978-966-305-064-5

Зміст

ТЕМА 1. Проектування системи керування на базі МП КР580ВМ80А.....	
1.1. Мікропроцесор КР580ВМ80А.....	
1.1.1. Умовне позначення та призначення виводів мікропроцесора.....	
1.1.2. Структура схеми мікропроцесора КР580ВМ80А.....	
1.1.3. Слово стана МП.....	
1.1.4. Робота мікропроцесора в режимі запису інформації.....	
1.1.5. Робота мікропроцесора в режимі читання.....	
1.1.6. Переривання в МП.....	
1.1.7. Прямий доступ до пам'яті.....	
1.1.8. Програмування та система команд мікропроцесора.....	
1.1.8.1. Мікропроцесорна система на основі МП К580ВМ80А з точки зору програмування.....	
1.1.8.2. Стек.....	
1.1.8.3. Класифікація команд МП.....	
1.2. Мікропроцесорний комплект КР580.....	
1.2.1. Тактовий генератор КР580ГФ24.....	
1.2.2. Системний контролер КР580ВК28.....	
1.2.3. Регістр КР580ІР82 (ІР83).....	
1.2.4. Шинний формувач КР580ВА86 (ВА87).....	
1.2.5. Контролер переривань КР580ВН59А.....	
1.2.6. Контролер прямого доступу до пам'яті КР580ВТ57.....	
1.2.7. Програмований таймер-лічильника КР580ВІ53.....	
1.3. Організація мікропроцесорної системи керування на основі мікропроцесорного комплекту КР580.....	
1.3.1. Організація клавіатури мікропроцесорної системи.....	
1.3.2. Організація пристрою відображення інформації.....	
1.3.3. Побудова технічних засобів для узгодження сигналів мікропроцесорної системи з об'єктами керування.....	
1.3.3.1. Вивід дискретних сигналів.....	

1.3.3.2. Ввід дискретних сигналів.....	
1.3.3.3. Ввід аналогових сигналів.....	
1.3.3.4. Виведення аналогових сигналів.....	
1.4. Контрольні запитання.....	
ТЕМА №2. Проектування системи керування на базі МП.....	
2.1. Мікропроцесор K1810BM86.....	
2.1.1. Умовне позначення мікропроцесора та загальна характеристика.....	
2.1.2. Призначення виводів МП.....	
2.1.3. Вибір режиму конфігурації.....	
2.1.4. Структурна схема мікросхеми.....	
2.1.5. Адресація портів вводу-виводу.....	
2.1.6. Адресація пам'яті.....	
2.1.7. Фізична організація пам'яті.....	
2.1.8. Переривання.....	
2.1.9. Часові діаграми роботи МП у мінімальному режимі роботи.....	
2.1.9.1. Часова діаграма читання пам'яті або портів вводу-виводу в мінімальному режимі роботи.....	
2.1.9.2. Часова діаграма запису у пам'ять або порти введення-виведення в мінімальному режимі роботи.....	
2.1.9.3. Часова діаграма підтвердження переривань у мінімальному режимі роботи.....	
2.1.9.4. Часова діаграма прямого доступу до пам'яті у мінімальному режимі роботи.....	
2.1.10. Часові діаграми роботи МП у максимальному режимі роботи.....	
2.1.10.1. Часова діаграма читання пам'яті або портів вводу-виводу в максимальному режимі роботи.....	
2.1.10.2. Часова діаграма запису пам'яті чи портів вводу-виводу в максимальному режимі роботи.....	
2.1.11. Початкова ініціалізація.....	
2.1.12. Система команд МП.....	
2.1.13. Мультипроцесорні системи.....	
2.2. Мікропроцесор K1810BM87.....	

2.3. Арифметичний співпроцесор K1810BM87.....	
2.3.1. Призначення виводів співпроцесора K1810BM87.....	
2.3.2. Робота співпроцесора K1810BM87.....	
2.4. Мікропроцесорний комплект K1810.....	
2.4.1. Структура мікропроцесорного комплекту.....	
2.4.2. Генератор тактових сигналів K1810ГФ24.....	
2.4.3. Контролер системної шини K1810ВГ88.....	
2.4.3.1. Режими роботи мікросхеми.....	
2.5. Організація мікропроцесорної системи керування на основі мікропроцесорного комплекту K1810.....	
2.5.1. Мінімальний режим.....	
2.5.2. максимальний режим.....	
2.5.3. Арбіт магістралі КР 1810ВБ89.....	
2.5.3.1. Призначення виводів мікросхеми.....	
2.6. Організація переривань мікропроцесорної системи на основі мікропроцесорного комплекту K1810.....	
2.6.1. Схеми реалізації протоколу переривань.....	
2.7. Контрольні запитання.....	
ТЕМА №3. Проектування системи керування на базі однокристальних мікроЕОМ КМ1816ВЕ48.....	
3.1. Призначення виводів та умовне позначення мікросхеми мікроЕОМ КМ1816ВЕ48.....	
3.2. Структура схеми мікросхеми мікроЕОМ КМ1816ВЕ48.....	
3.3. Використання зовнішньої пам'яті та розширеного вводу-виводу.....	
3.3.1. МК-системиз зовнішньою пам'яттю програм.....	
3.3.2. МК-система з зовнішньою пам'яттю даних.....	
3.3.3. МК-система з розширеним вводом/виводом.....	
3.4. Робота з клавіатурами.....	
3.4.1. Різновидності клавіатур.....	
3.4.2. Ввід коду натисненої клавіші.....	
3.4.3. Сканування.....	

3.4.4. Усунення деренчання контактів.....	
3.4.5. Чекання звільнення клавіші.....	
3.5.6. Ідентифікація натисненої клавіші.....	
3.4.7. Оформлення процедури вводу.....	
3.5. Виведення і відображення інформації. Індикатори.....	
3.5.1. Виведення символу на індикатори.....	
3.5.2. Виведення інформації на лінійний дисплей.....	
3.6. Введення інформації з датчиків.....	
3.6.1. Опитування двійникового датчика. Чекання події.....	
3.6.2. Чекання статичного сигналу.....	
3.6.3. Чекання імпульсивного сигналу.....	
3.6.4. Усунення деренчання контактів.....	
3.6.5. Підрахунок кількості імпульсів.....	
3.6.5.1. Підрахунок числа імпульсів між двома подіями.....	
3.6.5.2. Підрахунок числа імпульсів за заданий проміжок часу.....	
3.6.6. Опитування групи двійникових датчиків.....	
3.6.7. Опитування групи імпульсних датчиків.....	
3.7. Контрольні запитання.....	
ТЕМА №4. Проектування системи керування на базі однокристальних мікроЕОМ сімейства MCS-51.....	
4.1. Однокристальні мікроЕОМ сімейства MCS-51. Основні характеристики.....	
4.2. Архітектура ОМЕОМ 80С51.....	
4.2.1. Умовне позначення та призначення виводів мікроЕОМ.....	
4.3. МК-системи з зовнішньою пам'яттю програм.....	
4.4. Розширення ОЗП.....	
4.5. Ввід-вивід у МК-системах.....	
4.6. Лічильник-таймер у MCS-51.....	
4.6.1. Режими роботи таймера.....	
4.7. Послідовний інтерфейс.....	

4.7.1. Універсальний асинхронний приймопередавач.....	
4.7.2. Регістр керування/статусу УАПП.....	
4.7.3. Робота УАПП у мільтимікроконтролерних системах.....	
4.7.4. Швидкість послідовного обміну.....	
4.7.5. Особливості роботи УАПП у різних режимах.....	
4.7.5.1. Режим 0.....	
4.7.5.2. Режим 1.....	
4.7.5.3. Режим 2, 3.....	
4.8. Система переривань.....	
4.9. Контрольні запитання.....	
ТЕМА №5. Проектування системи керування на базі PIC16X8X.....	
5.1. Особливості контролерів PIC16X8X.....	
5.2. Призначення виводів та позначення мікросхеми.....	
5.3. Архітектура PIC16X8X.....	
5.4. Робота мікроЕОМ.....	
5.5. Структура та робота операційного блока.....	
5.6. Структура ПЗП програм.....	
5.7.Структура ОЗП.....	
5.7.1. Пряма та непряма адресація даних.....	
5.8. EEPROM даних PIC16X8X.....	
5.9. Регістри спеціальних функцій.....	
5.9.1. Регістр конфігурації (OPTION).....	
5.10. Порти вводу-виводу.....	
5.11. Модуль таймера PIC16X8X.....	
5.12. Організація переривань PIC16X8X.....	
5.13. Початкова ініціалізація та встановлення у початковий стан.....	
5.13.1. Джерела скидання.....	
5.13.2. Скидання при ввімкненні живлення.....	

5.13.3. Watcdog-таймер.....	
5.14. Режими зниженого енергоспоживання.....	
5.15. Генератор та синхронізація.....	
5.15.1. Типи генераторів.....	
5.15.2. Робота з кварцовим резонатором.....	
5.15.3. Синхронізація від зовнішніх джерел.....	
5.15.4. RC-генератор.....	
5.16. Конфігурація та захист.....	
5.16.1. Ідентифікаційний код.....	
5.16.2. Конфігураційне слово.....	
5.17. Система команд PIC-контролерів сіреї PIC16X8X.....	
5.18. Прилади застосування.....	
5.18.1. Універсальний таймер.....	
5.18.2. Частомір на PIC-контролері.....	
5.19. Контрольні запитання.....	
ТЕМА №6. МІКРОКОНТРОЛЕРИ ATMEGA32.....	
6.1. Характеристики ядра МК AVR.....	
6.2. Розміщення та призначення виводів МК ATMrga32.....	
6.3. Архітектура ATMrga32.....	
6.4. Функціонування конвеєра, цикл виконання команд мікроконтролера AVR.....	
6.5. Регістр стану – SREG.....	
6.6. Організація пам'яті даних ATMrga32.....	
6.6.1. Регістровий файл.....	
6.6.2. Регістри вводу-виводу.....	
6.6.3. Стек.....	
6.6.4. Пам'ять EEPROM.....	
6.7. Система команд мікроконтролера AVR.....	
6.7.1. Група команд пересилання даних.....	

6.7.2. Група команд арифметичних операцій та порівняння.....	
6.7.3. Група команд роботи із бітами. Група команд управління мікросхемою. Команди передачі керування.....	
6.8. Порти вводу-виводу.....	
6.9. Система переривань.....	
6.10. Лічильники-таймери мікросхеми.....	
6.10.1. попередні подільники таймерів-лічильників.....	
6.10.2. Восьмирозрядні таймери-лічильники 0 та 2.....	
6.10.2.1. Режими роботи таймерів 0 та 2.....	
6.10.2.2. Регістр, пов'язані з таймером-лічильником 0.....	
6.10.2.3. Регістр, пов'язані з таймером-лічильником 2.....	
6.10.2.4. Біти загальних регістрів, використовуваних таймерами 0 та 2.....	
6.10.3. 16-розрядний таймер-лічильник 1.....	
6.10.3.1. Режими роботи таймера 1.....	
6.10.3.2. Регістри таймера-лічильника 1.....	
6.10.3.3. Біти загальних регістрів, що використовує таймер 1.....	
6.10.4. Перевивання від таймерів.....	
6.11. Загальна характеристика послідовних інтерфейсів мікросхеми.....	
6.11.1. Послідовний порт SPI.....	
6.11.1.1. Загальний опис логіки роботи.....	
6.11.1.2. Конфігурації із багатьма веденими.....	
6.11.1.3. Інтерфейс SPI у мікросхемі ATmega32.....	
6.11.1.4. Регістри SPI.....	
6.11.2. Розширення портів вводу-виводу.....	
6.11.3. Універсальний синхронно-асинхронний послідовний приймопередавач USART.....	
6.11.3.1. Загальний опис.....	
6.11.3.2. Формат кадру.....	
6.11.3.3. Регістри USART.....	

6.11.3.4. Ініціалізація послудовного порту	
6.11.3.5. Процедури передачі та прийому даних.....	
6.11.3.6. Багатопроцесорні системи і передача 9 бітів.....	
6.12. АЦП мікроконтролера.....	
6.12.1. Загальна характеристика АЦП мікросхеми.....	
6.12.2. Робота АЦП.....	
6.12.3. Попередній подільник та частота перетворення АЦП.....	
6.12.4. Джерела сигналів АЦП.....	
6.12.5. Вибір опорної напруги.....	
6.12.6. Зниження шумів перетворення.....	
6.12.7. Результати перетворення.....	
6.12.8. Переривання, пов'язанні із АЦП.....	
6.12.9. Приклади роботи із АЦП.....	
6.13. Компаратор мікроконтролера.....	
6.13.1. Вхідні сигнали компаратора.....	
6.13.2. Переривання від аналогового компаратора.....	
6.13.3. Регістр контролю аналогового компаратора.....	
6.14. Контрольні запитання.....	
Список літератури.....	

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

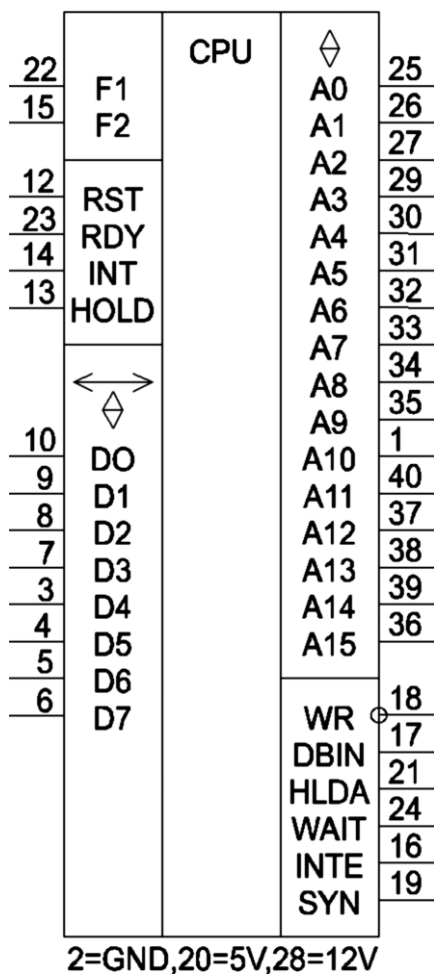


Рис. 1. Умовне позначення мікропроцесора

Призначення виводів мікропроцесора наведено в таблиці 2, умовне позначення – 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 запис 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

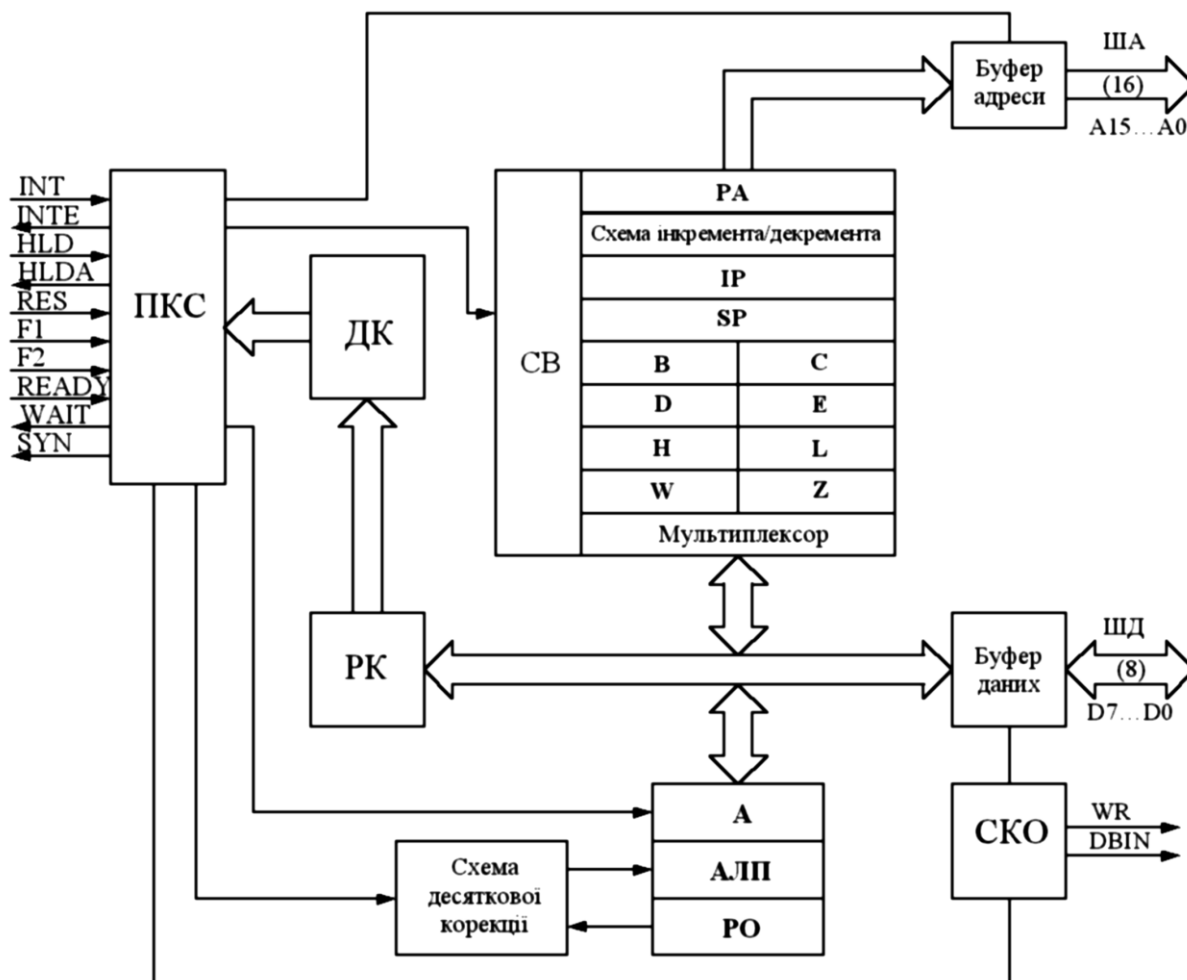


Рис. 2. Структурна схема мікропроцесора КР580ВМ80А

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 вміщується більшість результатів арифметичних і логічних команд
РО	Регістр ознак	Зберігає ознаки властивостей результату останньої арифметичної або логічної операції

Продовження табл. 3

1	2	3
СДК	Схема десяткової корекції	Блок призначений для перетворення результату роботи у 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	Програмно недоступні регістри, призначені для

	тимчасового зберігання	збереження другого і третього байтів команди
--	------------------------	--

Закінчення табл. 3

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:

- знака 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.

7							0
S	Z	x	AC	x	P	x	C

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 КР580ВМ80.

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	Запис у пам'ять	0	0	0	0	0	0	0	0
M4	Зчитування зі стека	1	0	0	0	0	1	1	0
M5	Запис у 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

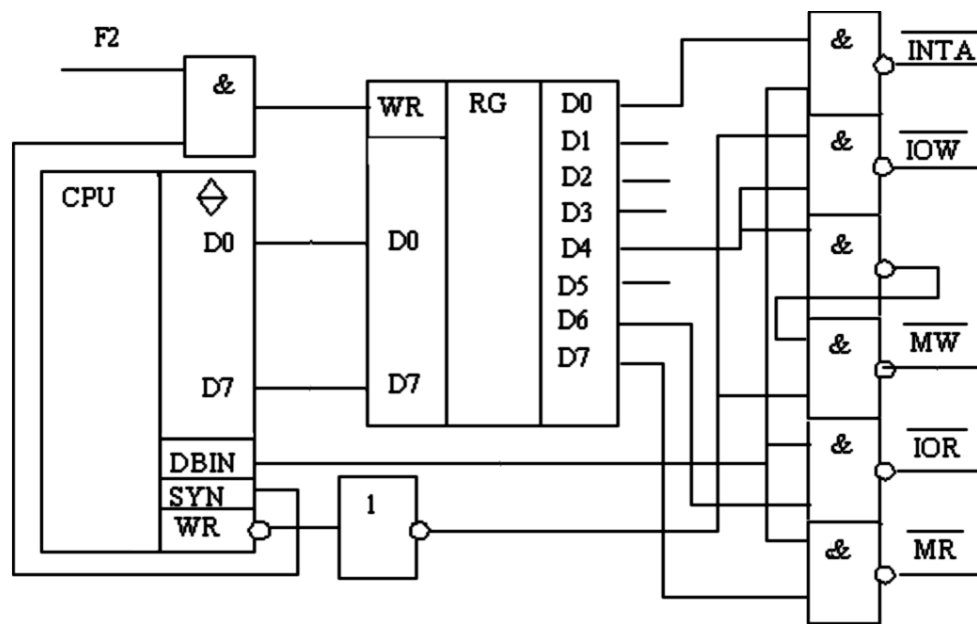


Рис. 4. Типова схема дешифрування слова стана

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

\overline{INTA} – сигнал підтвердження переривань.

\overline{MW} – запис 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} – запис у порти виводу OUT. При $\overline{IOW} = 0$ дозволяється запис даних у порт.

1.1.4. Робота мікропроцесора в режимі запису інформації

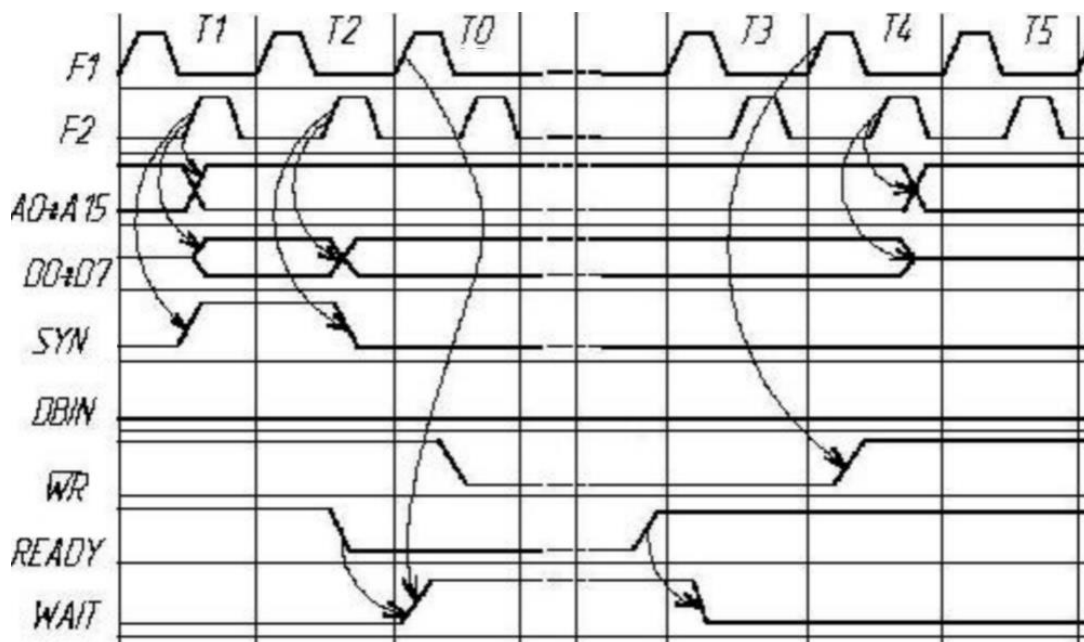


Рис. 5. Часова діаграма запису

Розглянемо часову діаграму роботи мікросхеми в режимі запису, 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 відбуватися запис інформації, а на шину даних виставляється код слова стана, що визначає тип обміну даними і вказує тип пристрою, куди буде відбуватись запис. 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 запис 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 протилежному випадку лінія \overline{WR} залишається 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 й дані. На цьому обмін по шині в циклі запису даних завершений.

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 відбуватися 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 зворотному випадку наступним тактом буде такт T3.

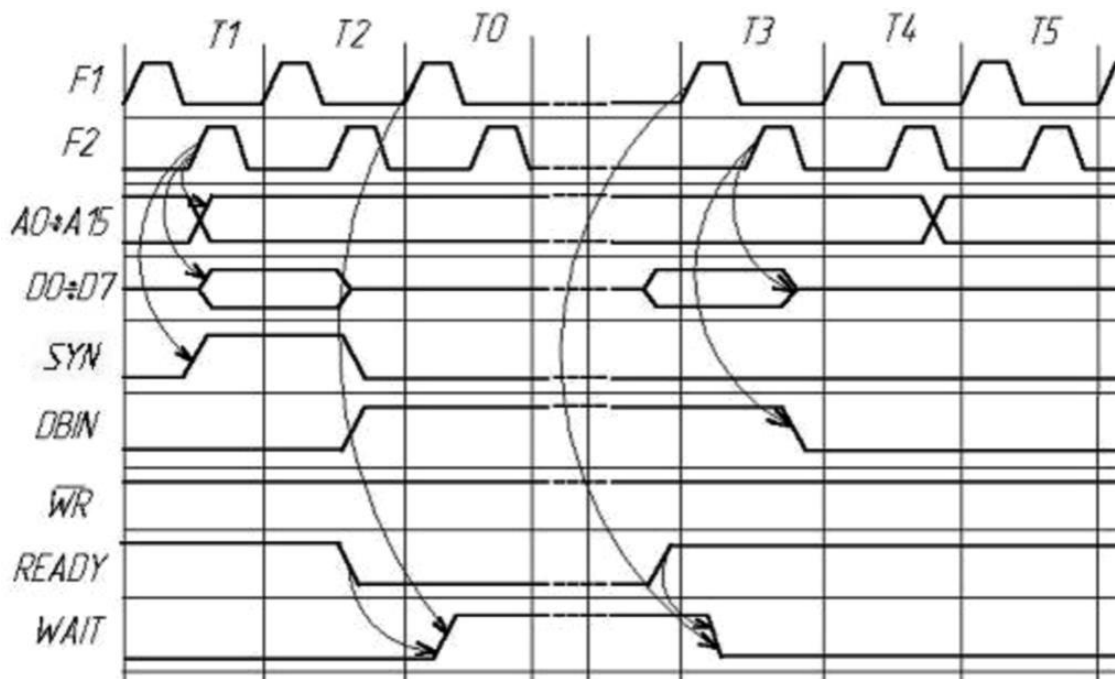


Рис. 6. Часова діаграма читання

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 тригера 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 шину 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 шину 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 закінчуватися 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 й у 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 К580ВМ80А з точки зору програмування

Структура мікропроцесорної системи на основі К580ВМ80А з точки зору програмування зображена на рис. 7. Мікропроцесор К580ВМ80А 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.

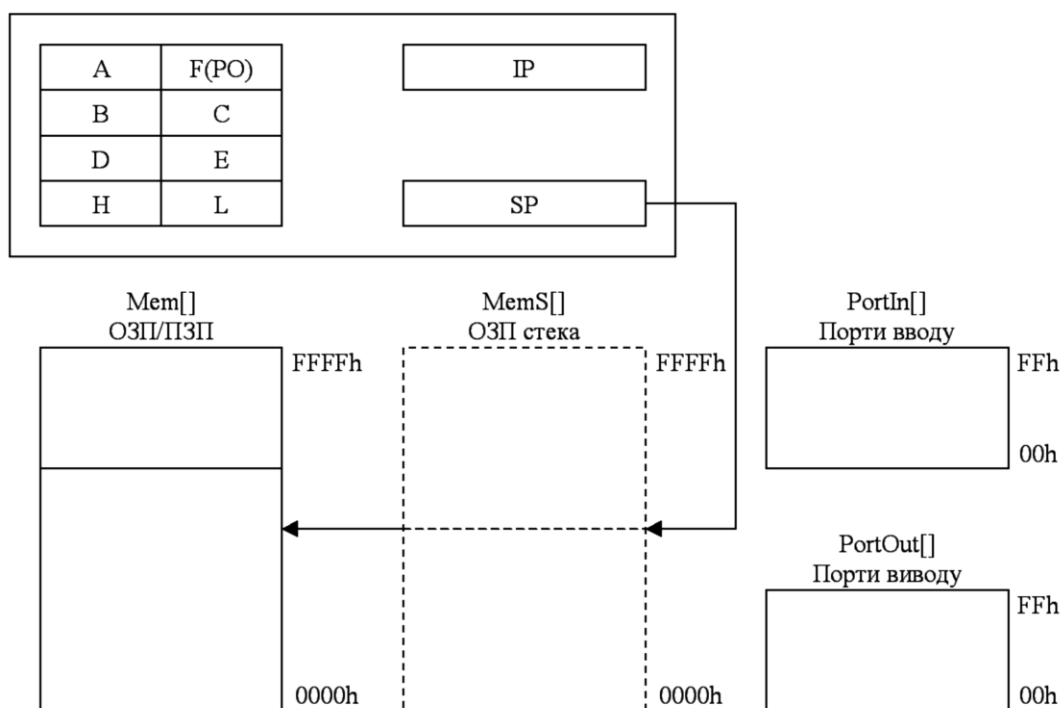


Рис. 7. Структура мікропроцесорної системи на основі K580BM80A з точки зору програмування

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 витягування даних дії відбуваються у 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 байтів);
- ◇ - трибайтні (код команди складається із 3 байтів).

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 поділяють на:

- 1) команди пересилання, котрі виконують пересилання даних 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, вліво та роботи з ознакою переносу;
- 6) команди переходу, які поділяють на команди: безумовного прямого й 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: вміщення 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 яких 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.

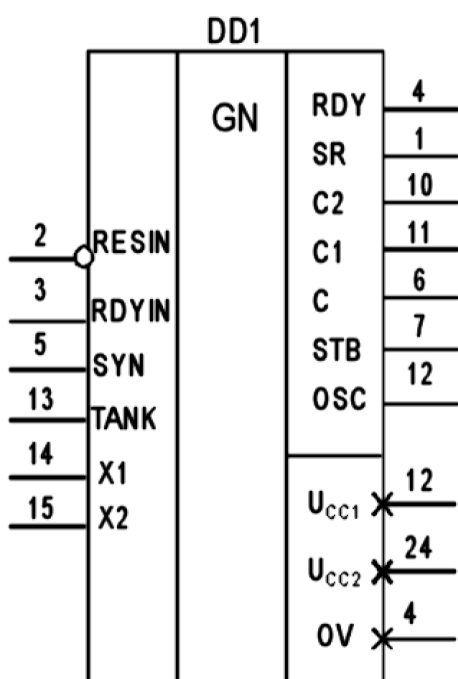
4. Контролери пристроїв та інтерфейсів:

- К580ВГ75 – контролер дисплея на основі електронно-променевої трубки;

- K580BB79 – контролер клавіатури та матричного дисплея;
- K580BK91 – контролер інтерфейсу КОП (IEEE-488);
- K580BG93 – шинний формувач для інтерфейсу КОП (IEEE-488).

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

1.2.1. 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 стана STB з періодом $T_{оп}/9$, де $T_{оп}$ – період тактових сигналів опорної частоти (частоти сигналів на лінії OSC);
- ◇ тактові імпульси (вивід C), які

Рис. 8. Тактовий генератор КР580ГФ24

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
RDY	Вихід	Вихідний сигнал готовності
SYN	Вхід	Сигнал синхронізації

1	2	3
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 контуру. 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 КР580ВК28

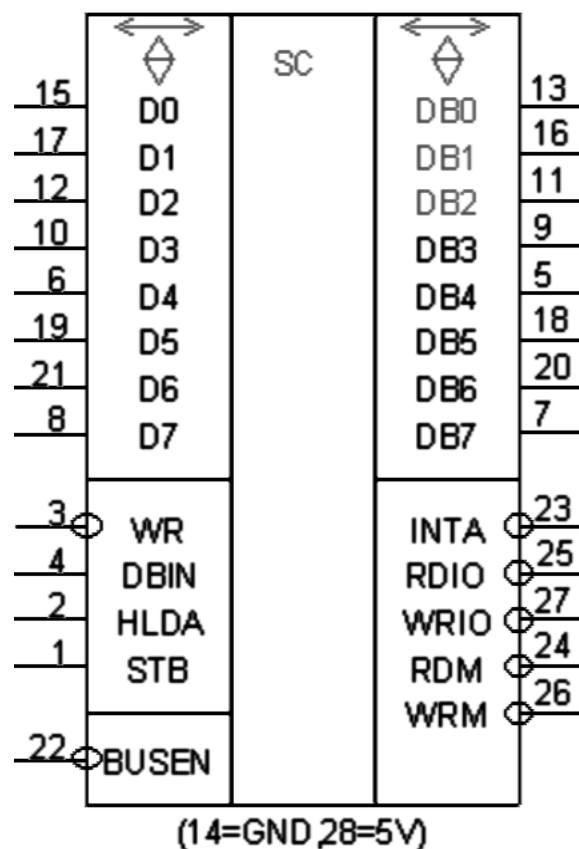


Рис. 9. Системний контролер КР580ВК28

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
DBIN	4	Вхід	Прийом інформації. Лінія DBIN мікропроцесора з'єднується з лінією DBIN контролера
DB0...DB7	13, 16, 11, 9, 5, 18, 20, 7	Входи-виходи	Двонаправлений канал даних системи
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
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
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	Запис в пам'ять. Логічний нуль на виході вказує, що мікропроцесор здійснює запис у пам'ять
WRIO	27	Вихід	Запис у ПБВ. Логічний нуль на виході вказує, що мікропроцесор здійснює запис у пристрій вводу-виводу

1	2	3	4
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)
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)

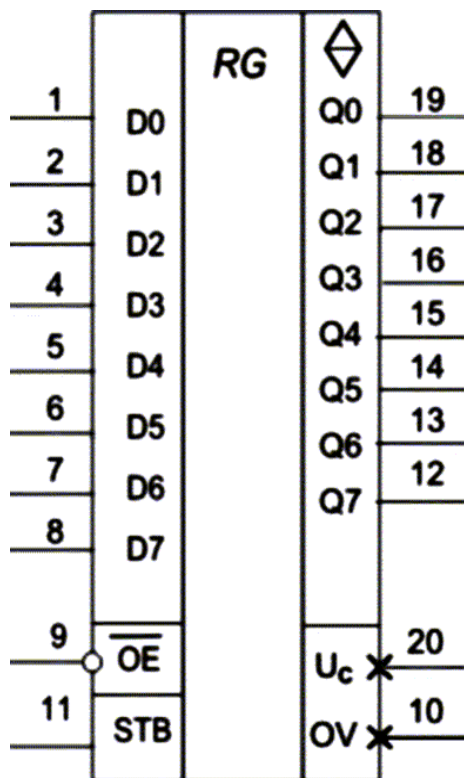


Рис. 10. Регістр КР580ІР82
(ІР83)

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.

Мікросхема КР580ІР83 має в порівнянні з КР580ІР82 інверсні виходи

$Q0...Q7$ (нумерація виводів обох мікросхем однакова).

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
Uc, 0V	20, 10	-	Входи живлення

1.2.4. Шинний формувач КР580ВА86 (ВА87)

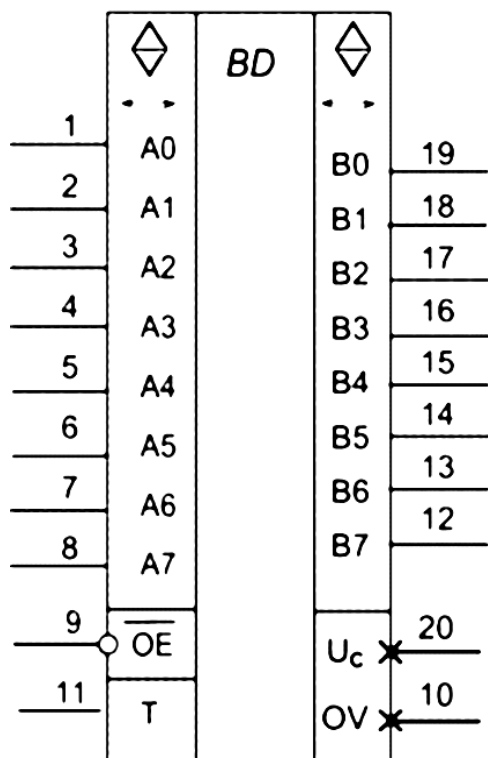


Рис. 11. Шинний формувач (прийомопередавач) КР580ВА86 (ВА87)

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 T змінює напрям передавання даних. Якщо T дорівнює лог. 0, забезпечується передавання даних з ліній B0...B7 до ліній A0...A7, у протилежному випадку – з ліній A0...A7 до ліній B0...B7. І в першому, і в другому випадках сигнал OE повинен бути в стані лог. 0, при OE=1 лінії шин A та B переводяться у z-стан.

Таблиця 8. Призначення виводів мікросхеми

Позначення	Номер виводу	Тип	Призначення
A0...A7	1-8	Вхід	Входи-виходи даних А
B0...B7	19-12	Вихід	Входи-виходи даних В
OE	9	Вхід	Вхід дозволу передавання даних
T	11	Вхід	Вхід напрямку передавання даних
Ucc, GND	20, 10	-	Входи живлення

Мікросхема КР580ВА87 має в порівнянні із КР580ВА86 інверсні виходи B0...B7 (нумерація виводів обох мікросхем однакова).

1.2.5. Контролер переривань KP580BH59A

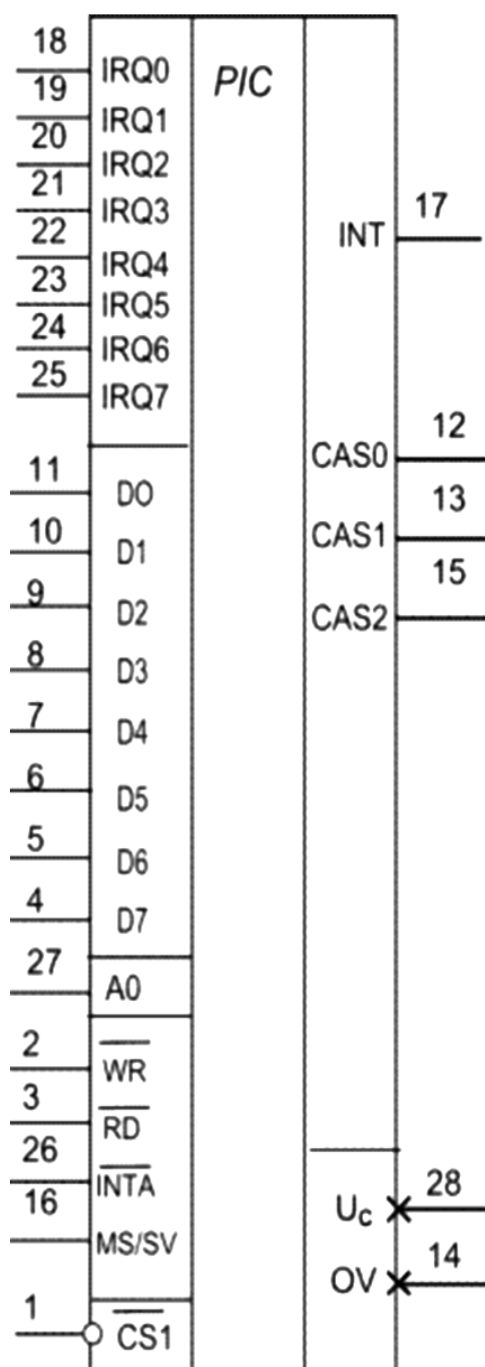


Рис. 12. Контролер переривань KP580BH59A

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 посиляється 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 запис даних у 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
Uc,	28	-	Напруга живлення
0V	14	-	Спільний вивід

Режим роботи контролера встановлюється програмним шляхом: здійснюється подача двох або трьох команд. У першій команді задаються значення розрядів A5...A7 молодшого байта адреси та задається режим роботи мікросхеми. У другій команді задаються значення старшого байта адреси. Найвищий пріоритет за замовчуванням має вхід IRQ0, найнижчий – IRQ7. Пріоритет входів IRQ може змінюватися програмно.

1.2.6. Контролер прямого доступу до пам'яті KP580BT57

Для обслуговування швидкодіючих зовнішніх пристроїв і збільшення пропускної 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 з можливістю визначення будь-якої з 64 Кбайт початкових адрес.

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

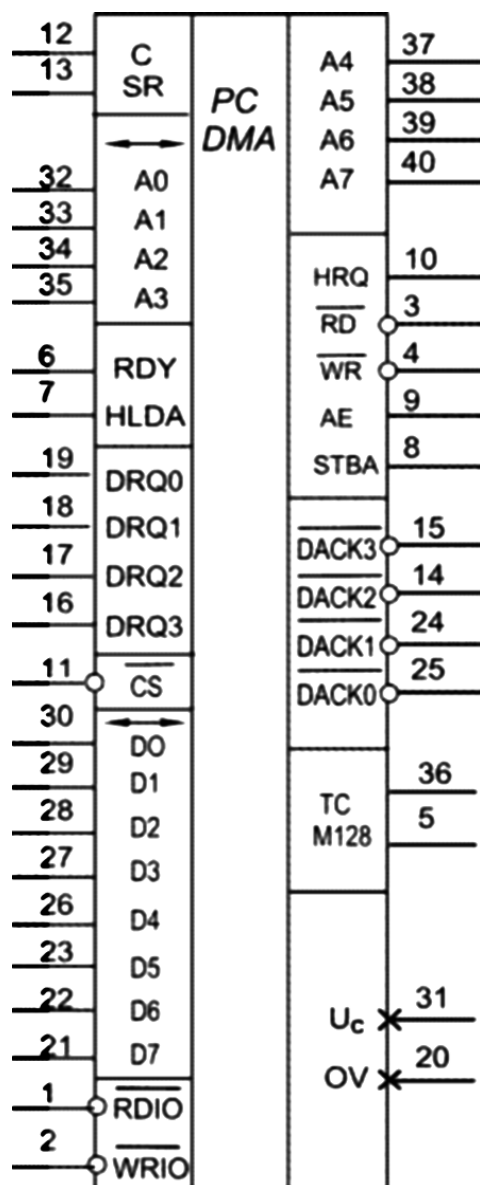


Рис. 13. Контролер прямого доступу до пам'яті (ПДП) KP580BT57

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
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
M128	5	Вихід	Модуль 128. Є активним, коли поточний цикл обміну є 128, або кратним 128 від кінця масиву даних

1	2	3	4
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
SR	13	Вхід	Сигнал початкового встановлення. Логічна 1 на лінії призводить до скидання мікросхеми
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
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 вказування, 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 регістр. 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
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 запис у пам'ять (зчитування з ПВВ). При цьому сигнал зчитування випереджає сигнал запису в пам'ять. В інших режимах роботи сигнал служить входом і використовується при читанні регістрів мікросхеми (активний рівень сигналу – нульовий)

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
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
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 регістрів 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).
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

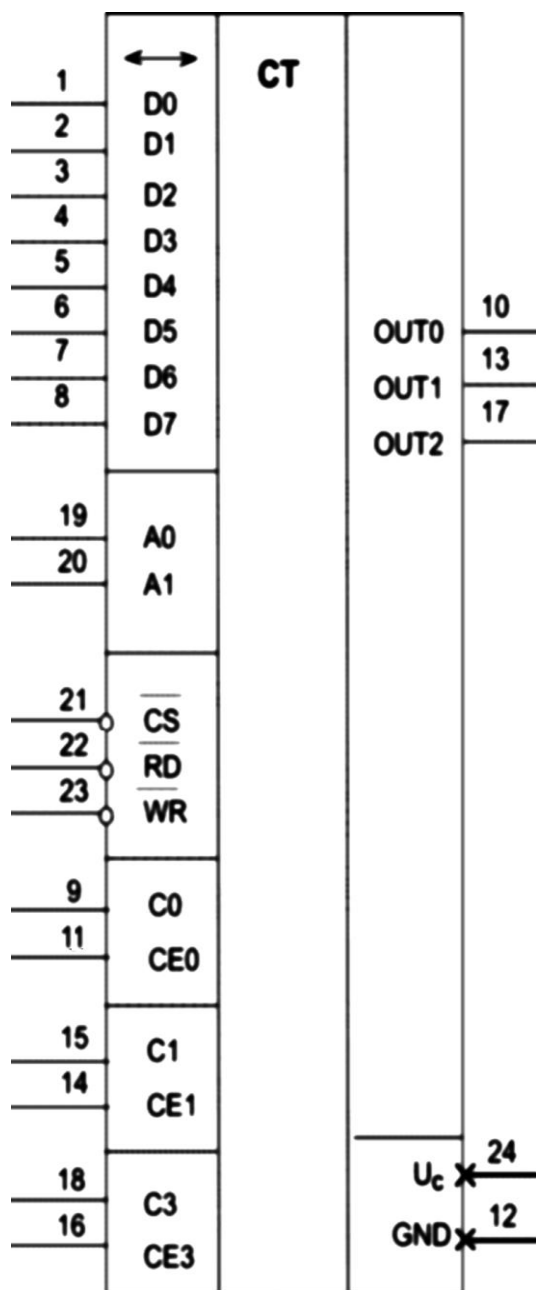


Рис. 14. Програмований таймер-лічильник КР580ВІ53

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

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 – 216;
- ◇ у десятковому коді – 104;

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 записі керуючого слова в

Позначення	Номер виводу	Тип виводу	Призначення
1	2	3	4
D0...D7	1-8	Входи/ виходи	Канал даних. По лініях передаються дані про стан мікросхеми та лічильників, а також здійснюється її програмування

Закінчення таблиці 11

1	2	3	4
A0, A1	19, 20	Входи	Адресні лінії. Лінії вказують номер лічильника або допоміжних регістрів (якщо A1A0 = 11), що програмуються або зчитуються в циклі обміну
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
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 про 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

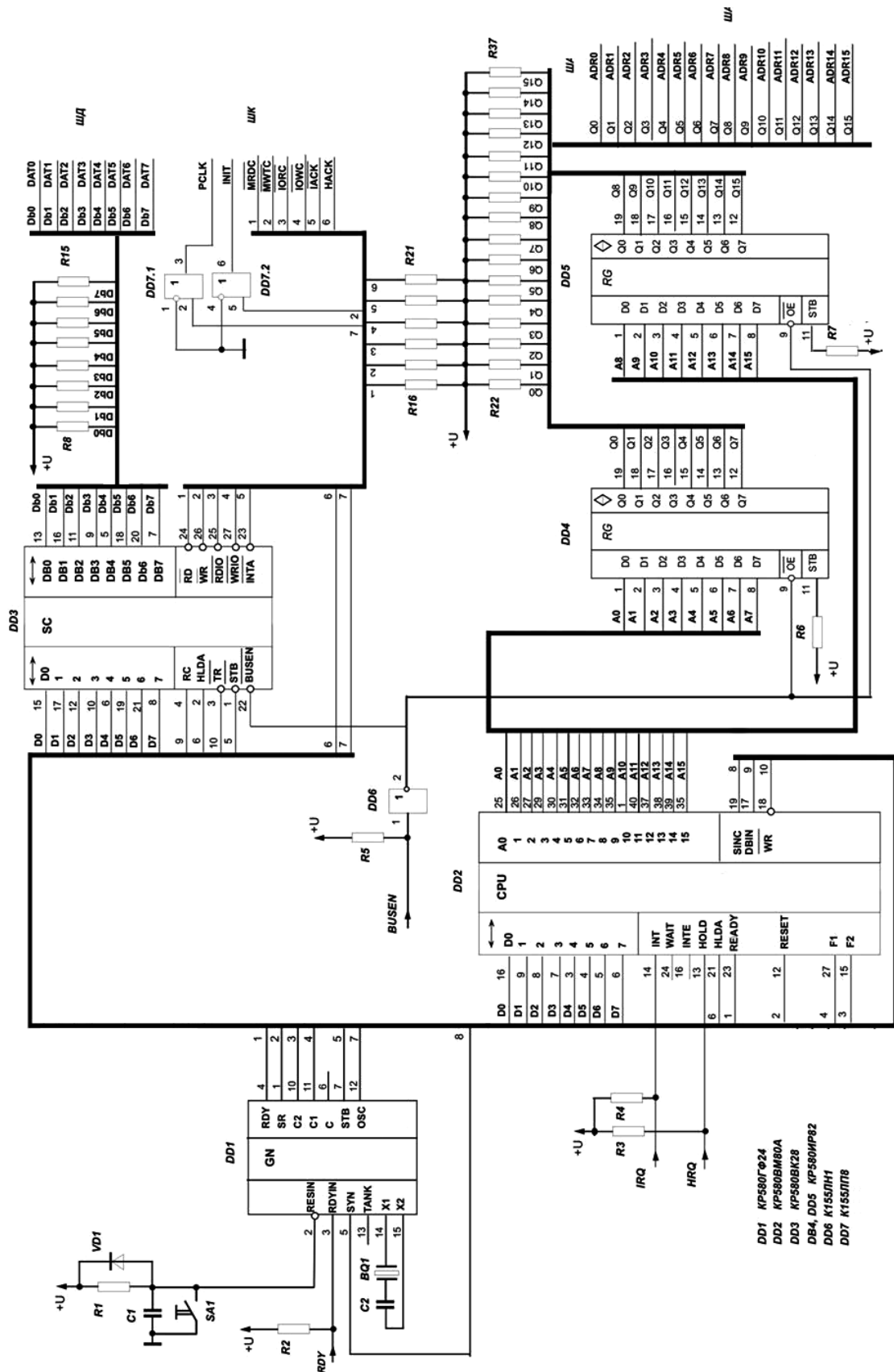
1.3. Організація мікропроцесорної системи керування на основі мікропроцесорного комплекту КР580

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 КР580ІР82, на яких побудована адресна шина ША (ADR0...ADR15).



- DD1 КР580ІР24
- DD2 КР580ІР80А
- DD3 КР580ІР28
- DD4, DD5 КР580ІР82
- DD6 К155ІН1
- DD7 К155ІП8

Рис. 15. Процесорний блок

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

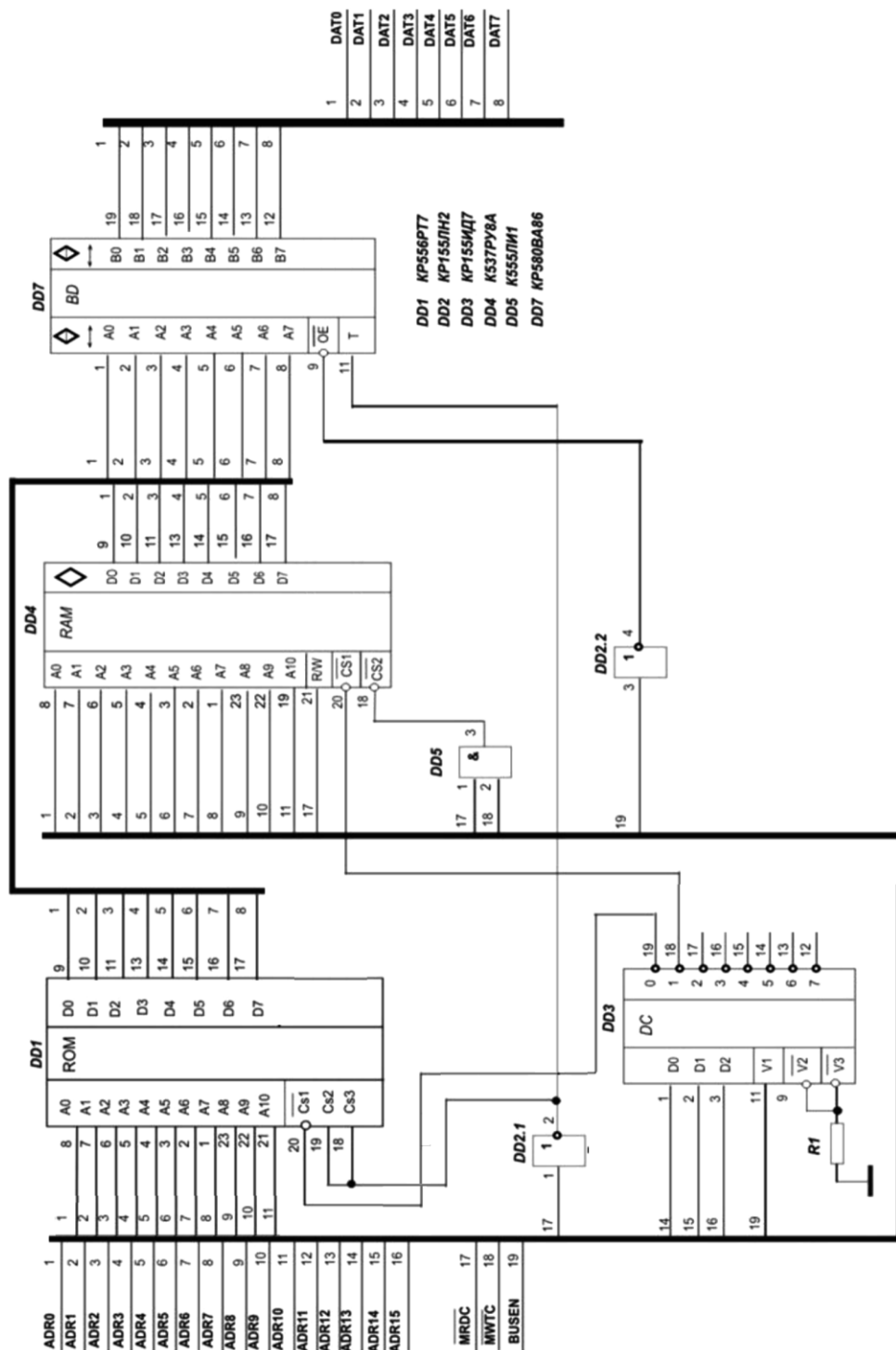


Рис. 16. Блок пам'яті

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

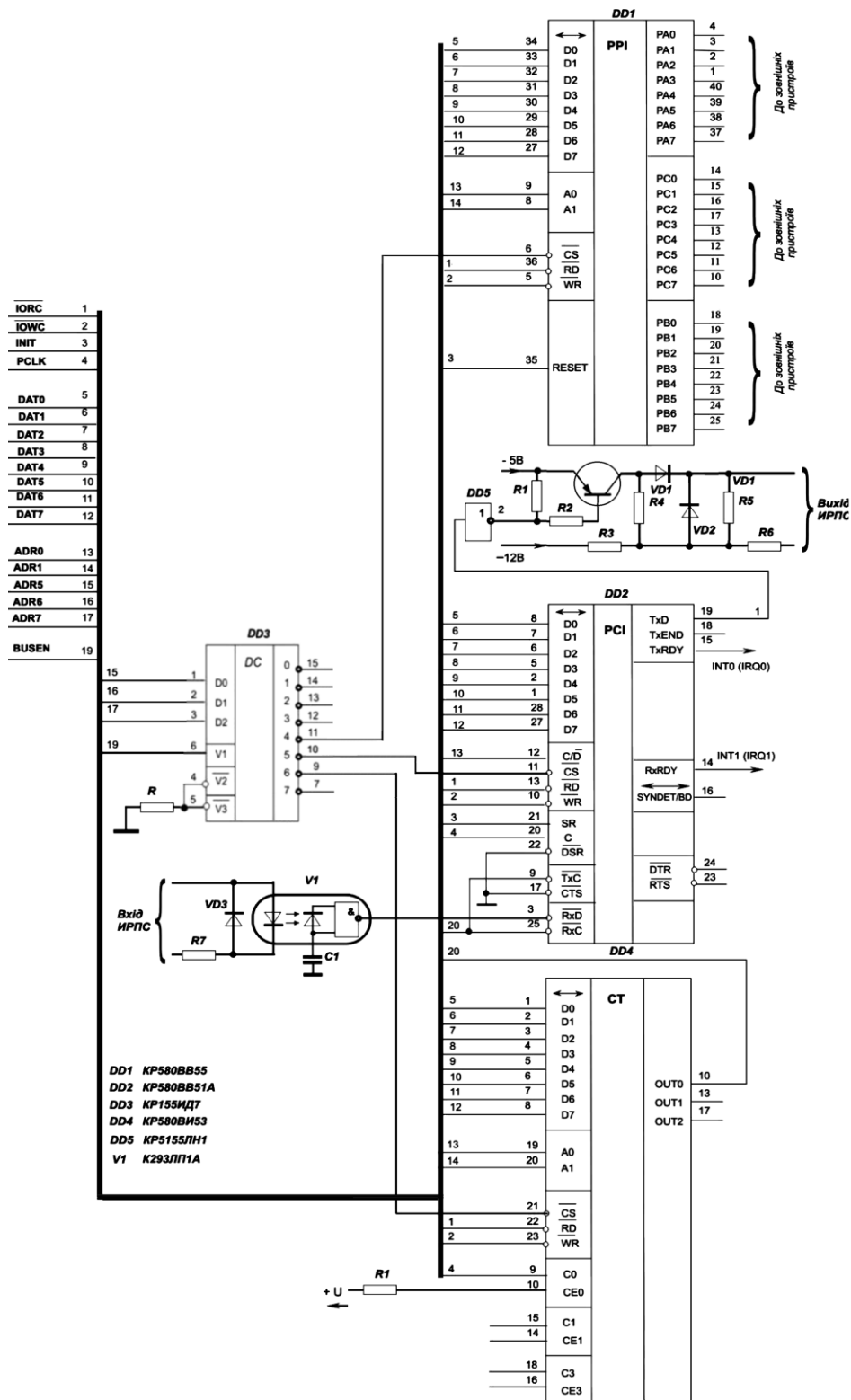


Рис. 17. Блок вводу-выводу

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.

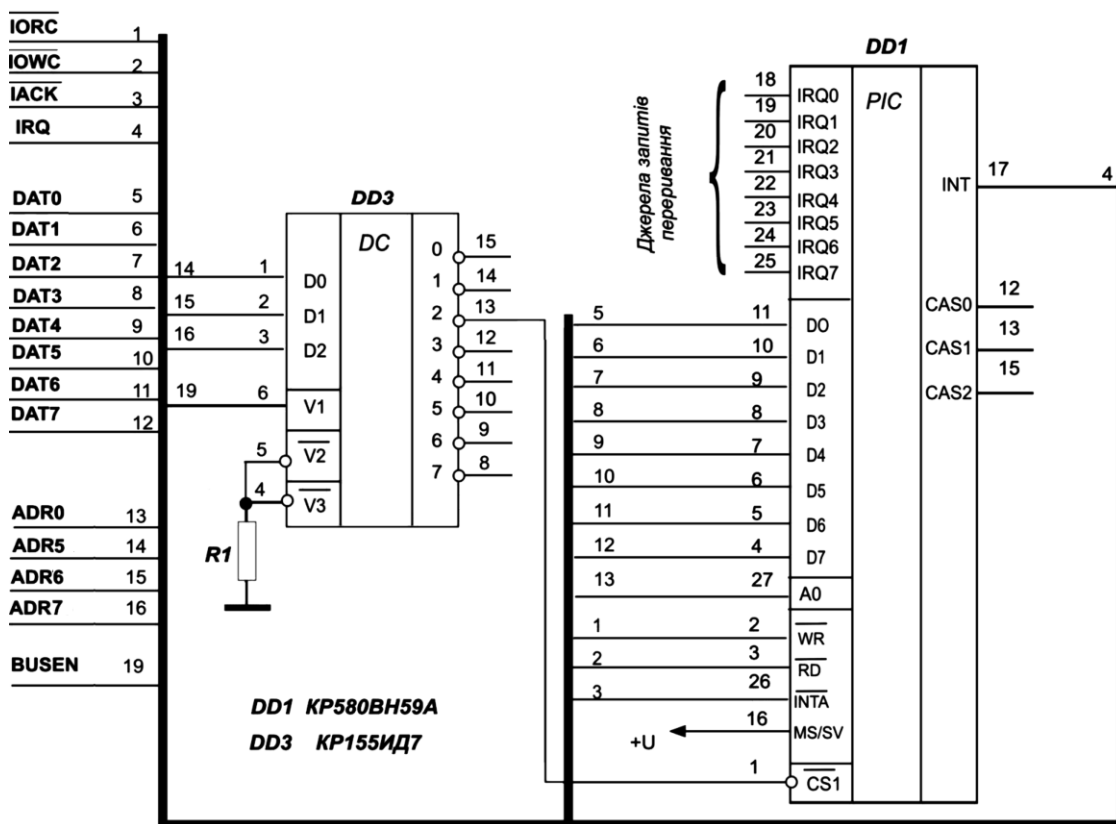


Рис. 18. Контролер переривань

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).

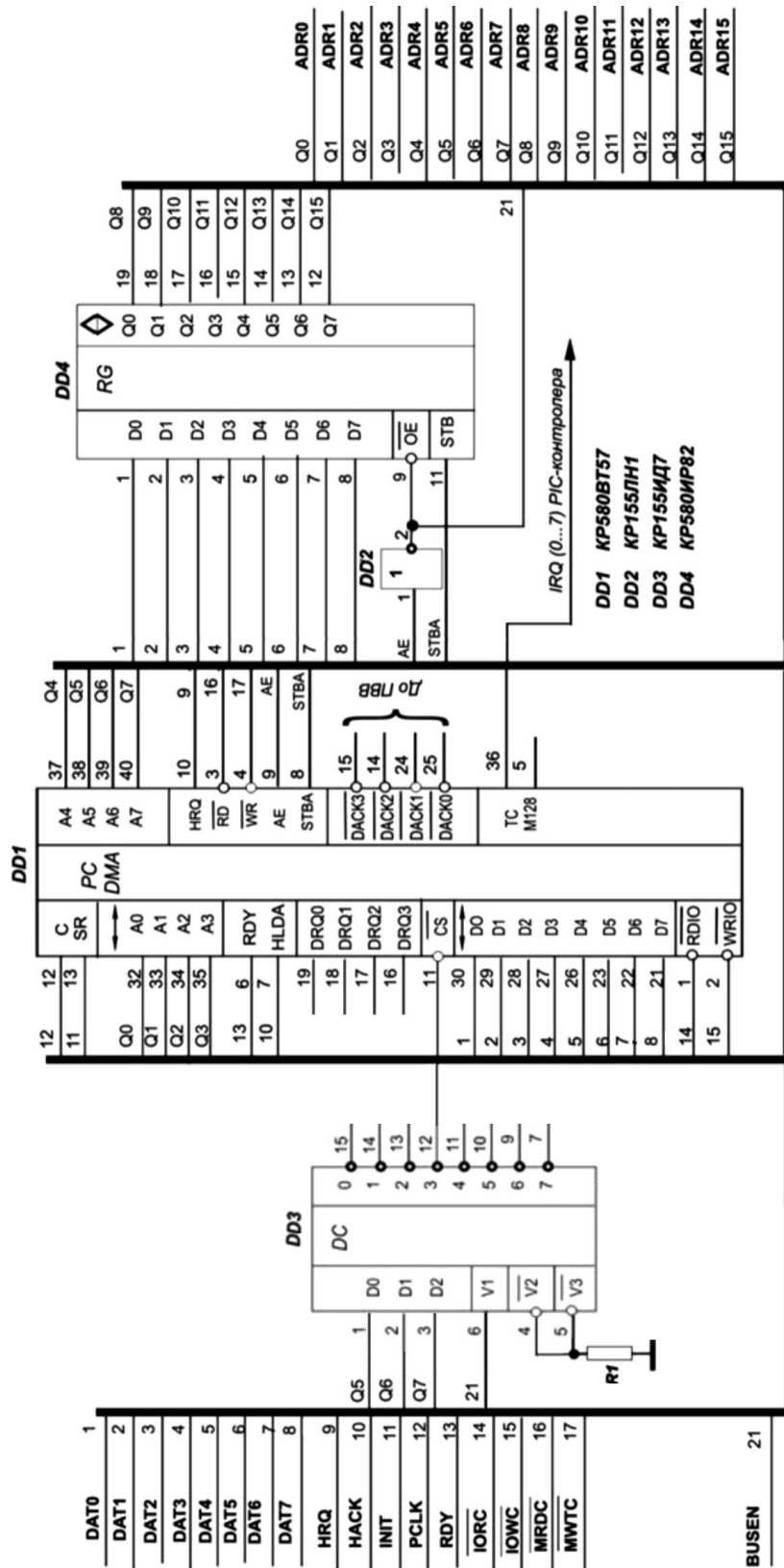


Рис. 19. Контролер прямого доступу до пам'яті

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 введення інформації на основі матриці ключів 5x5.

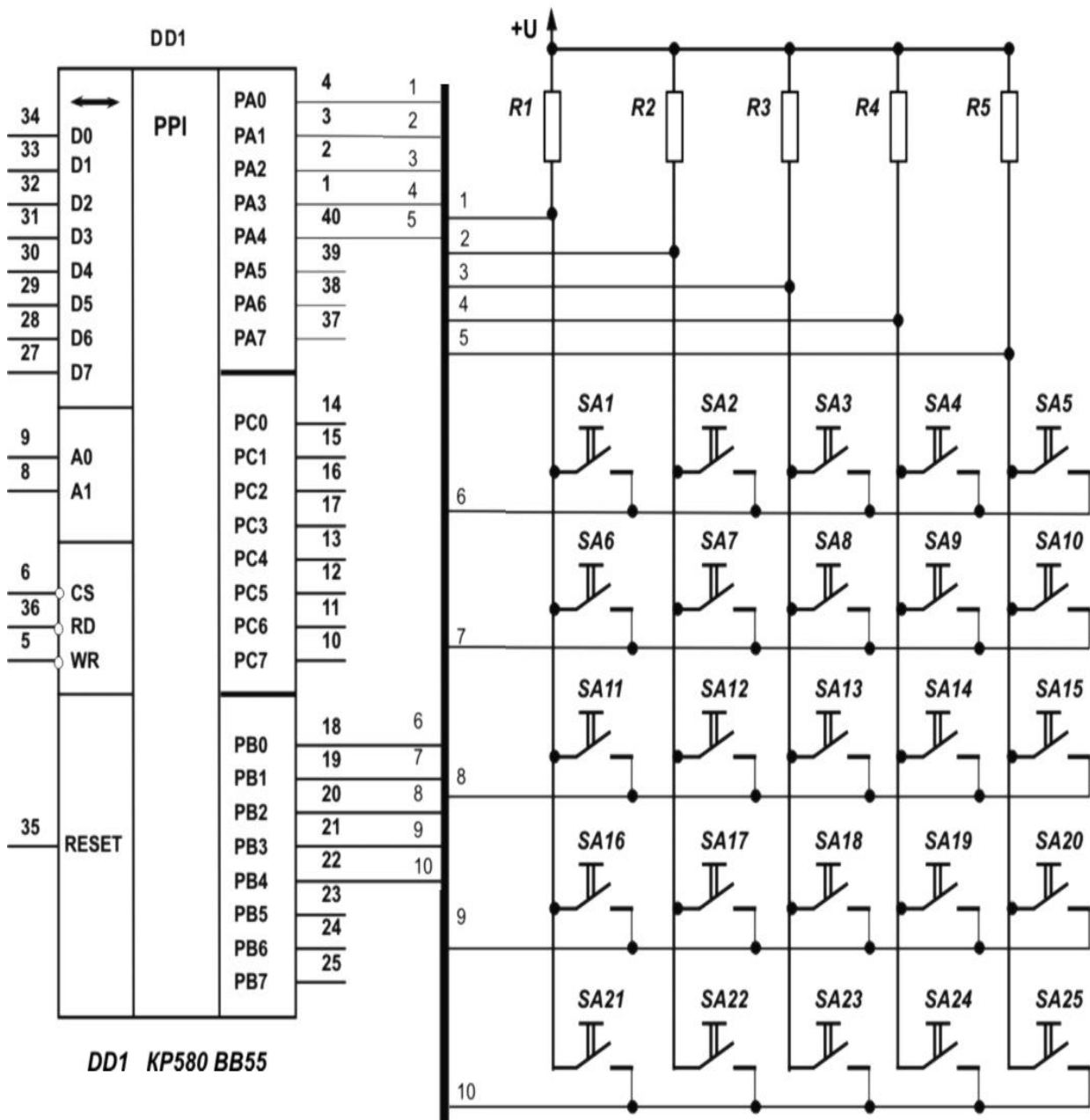


Рис. 20. Клавіатура на основі K580BB55

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

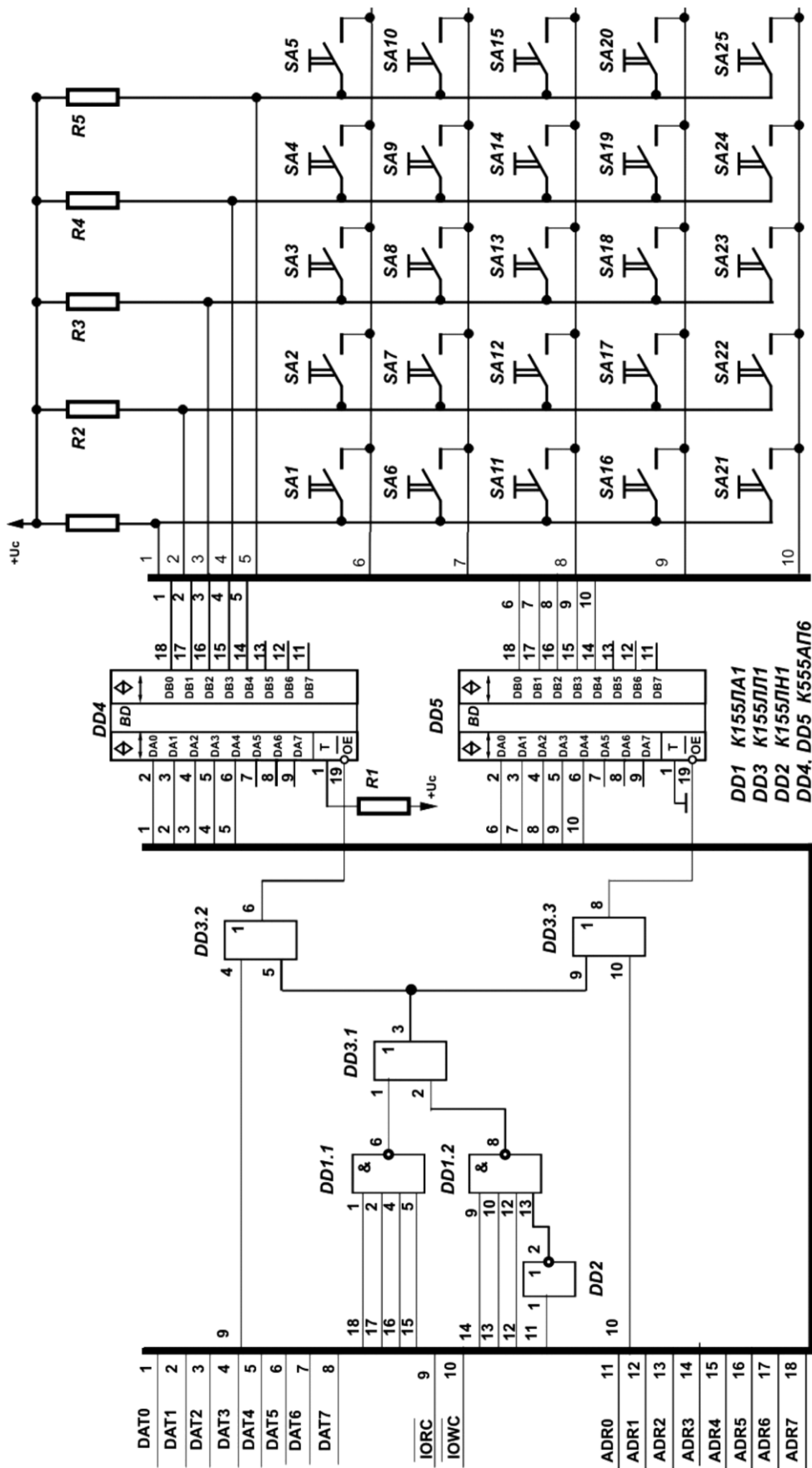


Рис. 21. Клавiатура

Ключі клавiшних матриць обох рисункiв розташованi так:

- ◇ один вивiд кожного нормально розiмкненого ключа з'єднаний з однiєю iз вертикальних лiнiй матриць, iнший – з однiєю iз горизонтальних лiнiй;
- ◇ горизонтальнi лiнiї матриць є її вхiдними лiнiями, вертикальнi – вихiдними. В кожен момент часу лише одна з горизонтальних лiнiй повинна мати рiвень логiчного “0”, всi iншi – рiвень логiчної “1”;
- ◇ коли на одну iз горизонтальних лiнiй подається “0”, стан вертикальних лiнiй перевiряється мiкропроцесором. Якщо при цьому вiдбувається замикання ключа з тих, що пiд'єднанi до цiєї горизонтальної лiнiї, то вiдповiдна йому вертикальна лiнiя також отримує рiвень “0”;
- ◇ за допомогою вiдповiдного програмного забезпечення мiкропроцесорна система встановлює почергово рiвень “0” на горизонтальних лiнiях матриць, перевiряючи при цьому стан її вертикальних лiнiй. Номери горизонтальної та вертикальної лiнiй матриць, якi мають рiвень “0”, i будуть визначати координати (номер) замкненого ключа.

У схемi на рис. 20 порт А, запрограмований на введення, здiйснює опитування вертикальних лiнiй матриць, а порт В, запрограмований на виведення, встановлює почергово на горизонтальних лiнiях матриць рiвень “0”. Для адресацiї портiв використовуються лiнiї ADR0 та ADR1 адресної шини мiкропроцесорної системи.

У схемi на рис. 21 опитування лiнiй матриць здiйснюється через прийомопередавач DD4 (працює на введення даних), а встановлення “0” на горизонтальних лiнiях матриць – через прийомопередавач DD5 (працює на виведення даних). Iншi мiкросхеми забезпечують адресацiю клавiатури (фiксована адреса FE), а також формують керуючi сигнали зчитування та запису.

Часто в якостi контролера клавiатури з органiзацiєю клавiатурної матриць 8x8 використовується запрограмований iнтерфейс KP580BB79, який складається з двох функцiонально незалежних частин – клавiатурної та дисплейної.

1.3.2. Організація пристрою відображення інформації

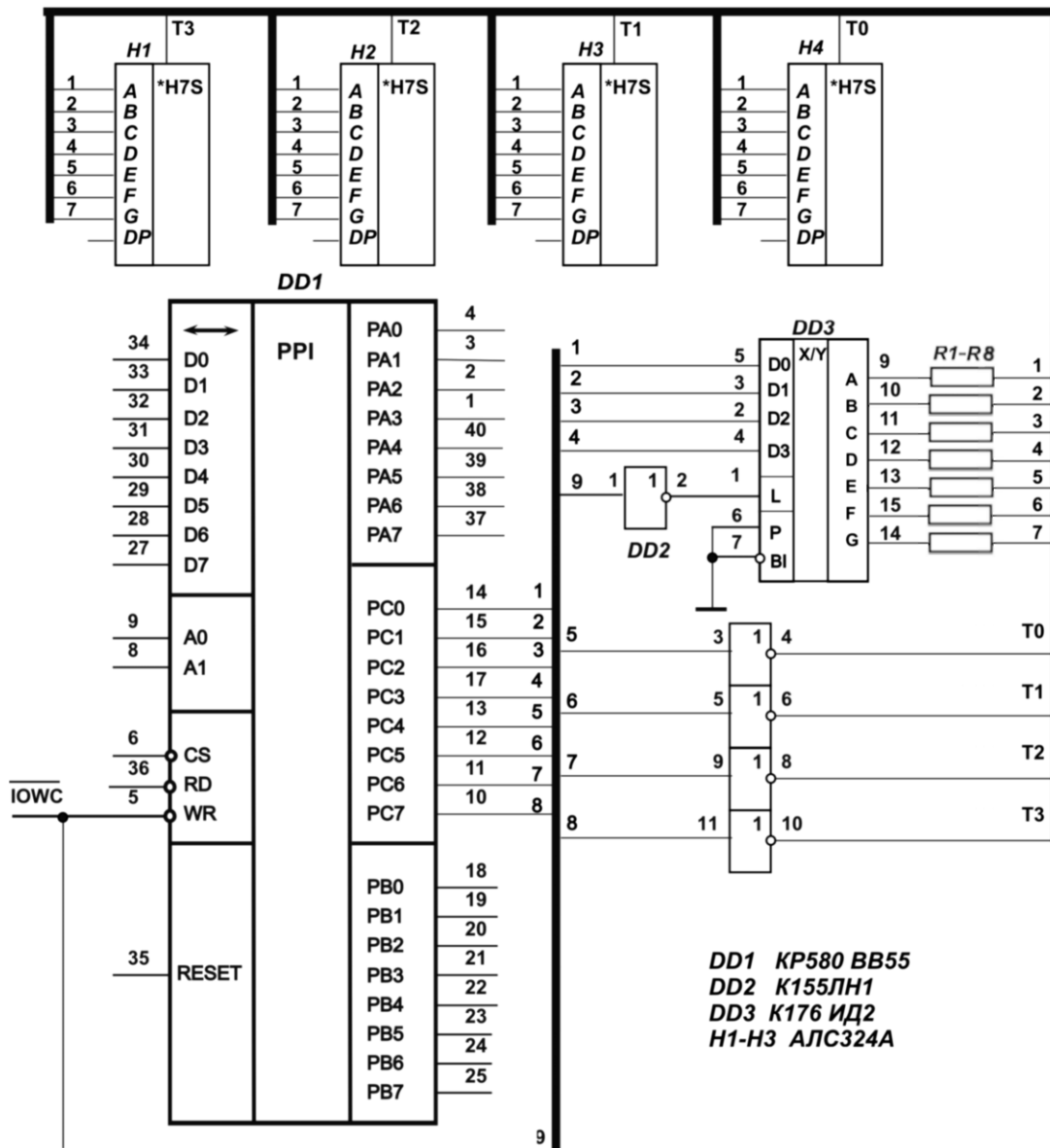


Рис. 22. Дисплей

На рис. 22 зображено один із можливих варіантів організації цифрового табло для відображення чотирьох десяткових цифр за допомогою світлодіодних семисегментних індикаторів АЛС324 А. Для виведення даних на індикатори у вигляді двійково-десятькового коду цифри (виводи С0...С3) та для визначення розряду цифри (виводи С4...С5) використовується програмований паралельний інтерфейс КР580ВВ55. Для перетворення двійково-десятькового 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 виведення 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 виведення інформації на алфавітно-цифрові дисплеї використовується контролер електронно-променевиx 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, пускачі й 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. 23а,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. 23в,г), вхідні ланцюги яких не

мають гальванічного зв'язку із мікропроцесорною системою керування, а також оптоелектронні тиристорні пари, що дозволяють комутувати малопотужні (рис. 23е) та силові кола змінного струму (рис. 23д).

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

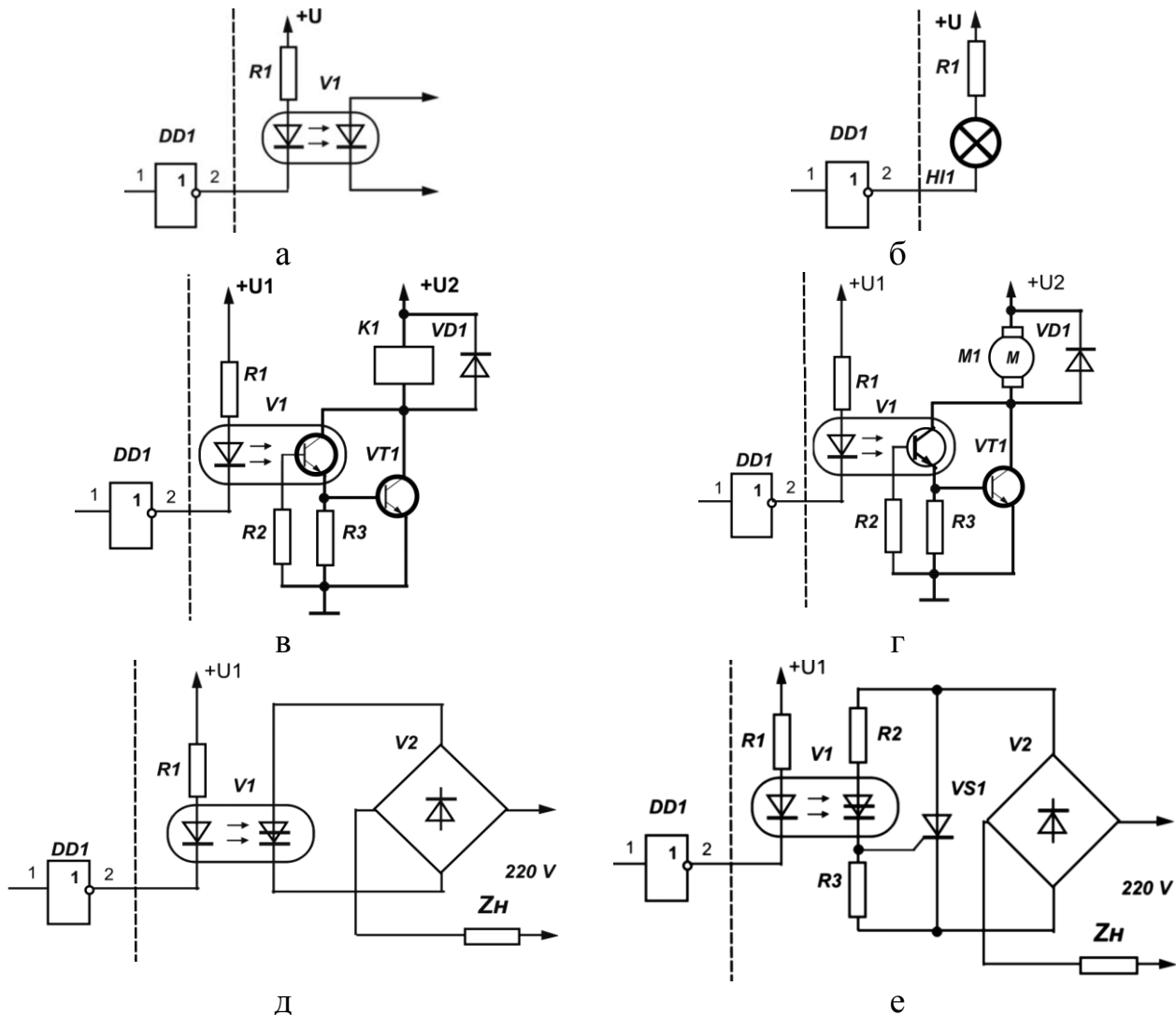


Рис. 23. Вихідні цифрові пристрої

Керування вихідними формувачами дискретних сигналів здійснюється за допомогою пристроїв виведення 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 датчиків дискретних сигналів у сигнали стандартної амплітуди (наприклад, ТТЛ рівня) для вводу в МП-систему.

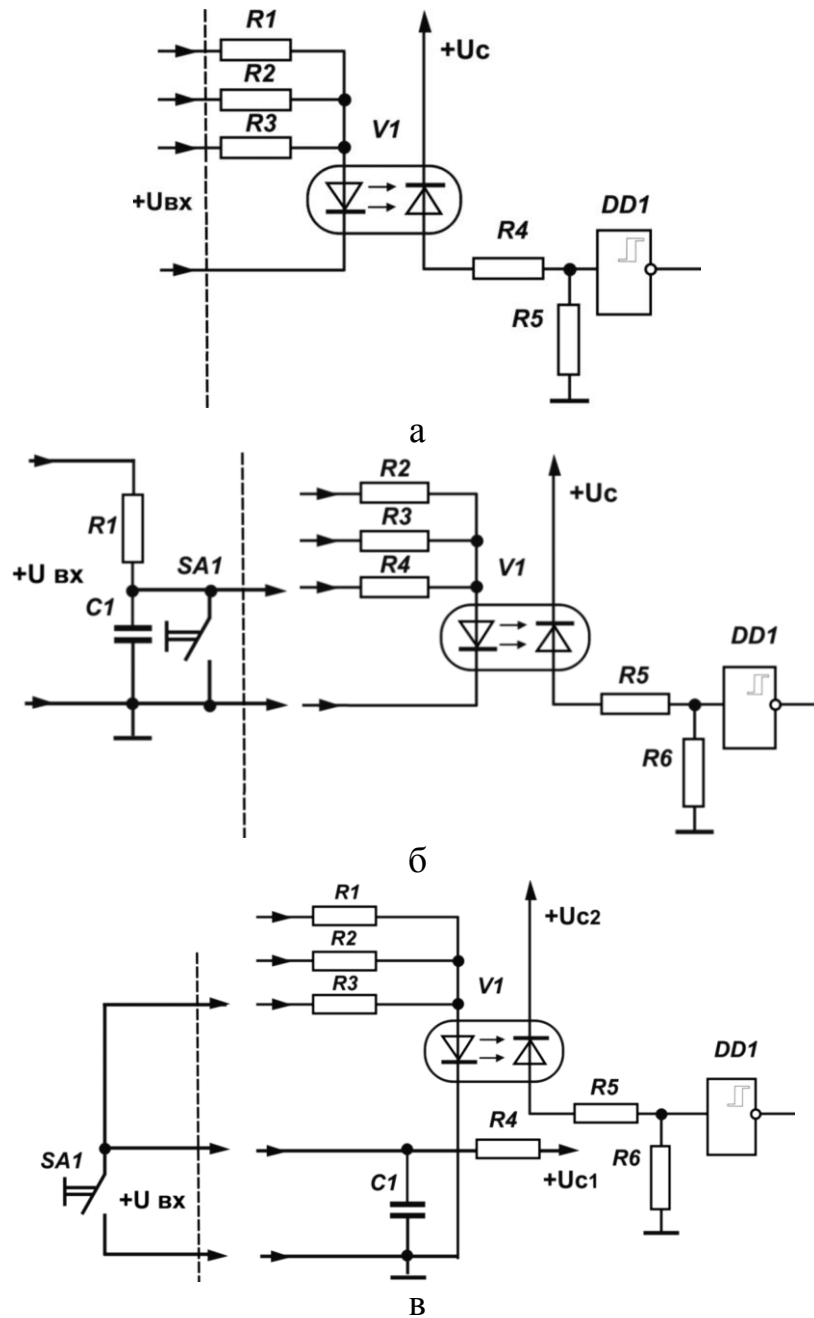


Рис. 24. Ввід дискретних сигналів

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 (наприклад, стан кінцевих вимикачів);
- ◇ кодовий, коли сукупність двійкових розрядів формує функційно пов'язану групу сигналів (код). Наприклад, код, котрий вказує кут повороту;
- ◇ імпульсний, за якого стан об'єкта визначається через кількість імпульсів. Наприклад, кількість деталей на конвеєрі відповідає кількості імпульсів від фотоелемента.

Типові схеми під'єднання дискретних давачів зображені на рис. 24а,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. 25а):

- ◇ комутатор аналогових сигналів (АК);

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

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

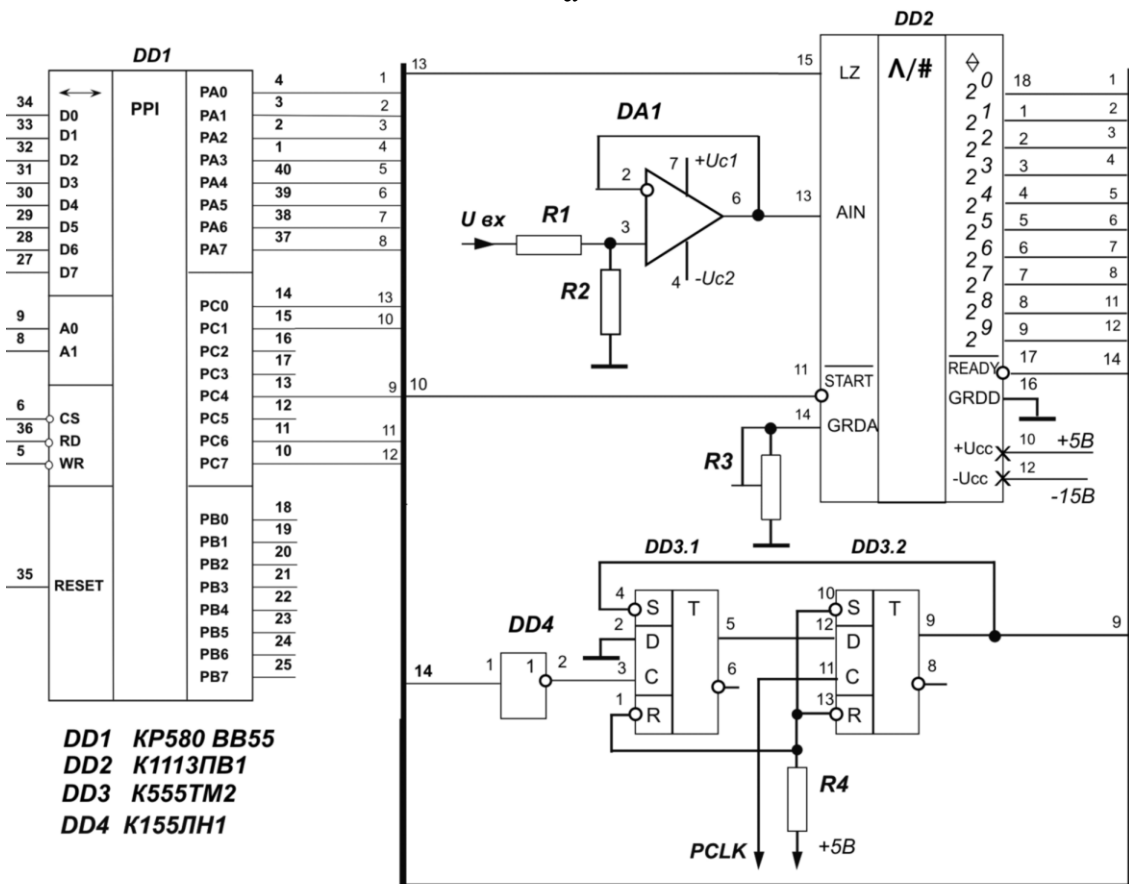
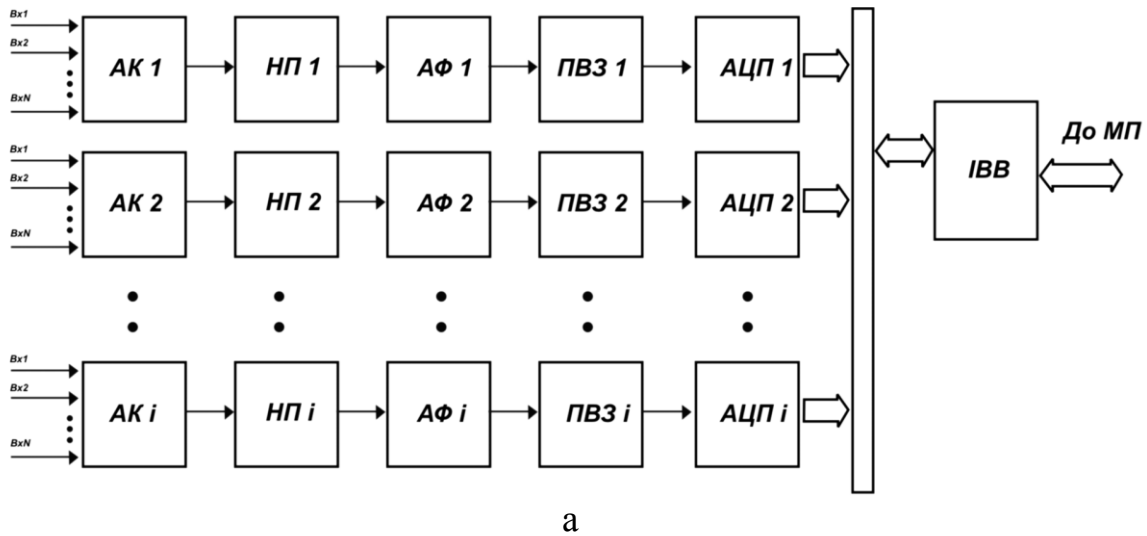


Рис. 25. Ввід аналогових сигналів

Аналоговий комутатор призначений для під'єднання аналогових давачів до входу підсилювача НП, який в свою чергу здійснює операцію пропорційного підсилення або послаблення (нормування) аналогових сигналів, що надходять від давачів, 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 мультиплекси 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 у двох режимах:

◇ вибірки, основним параметром якого є час вибірки T_b ;

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

АЦП залежно від застосування, поділяють 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. 25б зображена схема під'єднання АЦП К1113ПВ1А до джерела аналогового сигналу через нормуючий пристрій (К140УД7) та узгодження його з мікропроцесорним пристроєм через програмований паралельний інтерфейс

KP580BB55. АЦП K1113PB1A 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 й інформація з виходу АЦП через лінії PA0...PA7 та PC6, PC7 інтерфейсу DD1 надходить у мікропроцесор.

1.3.3.4. Виведення аналогових сигналів

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

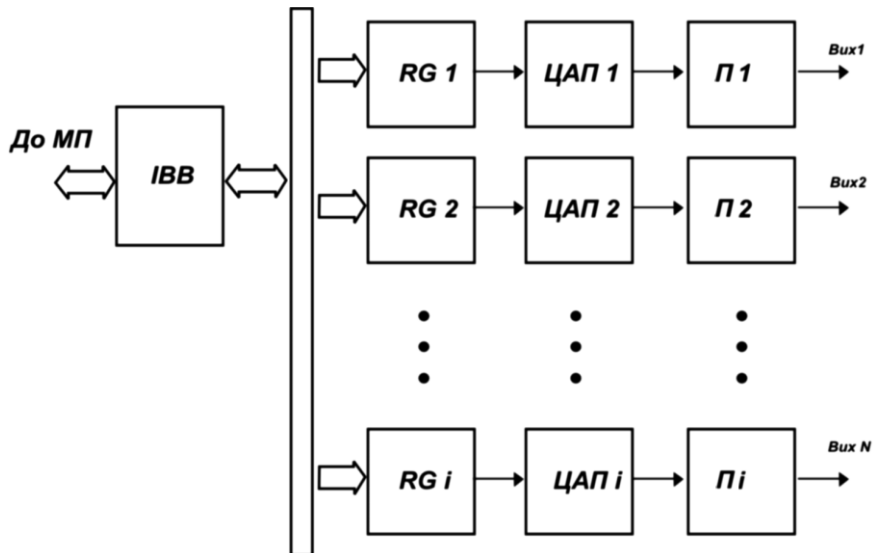
- ◇ буферні регістри (RG);
- ◇ цифроаналогові перетворювачі (ЦАП);
- ◇ підсилювачі (П).

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

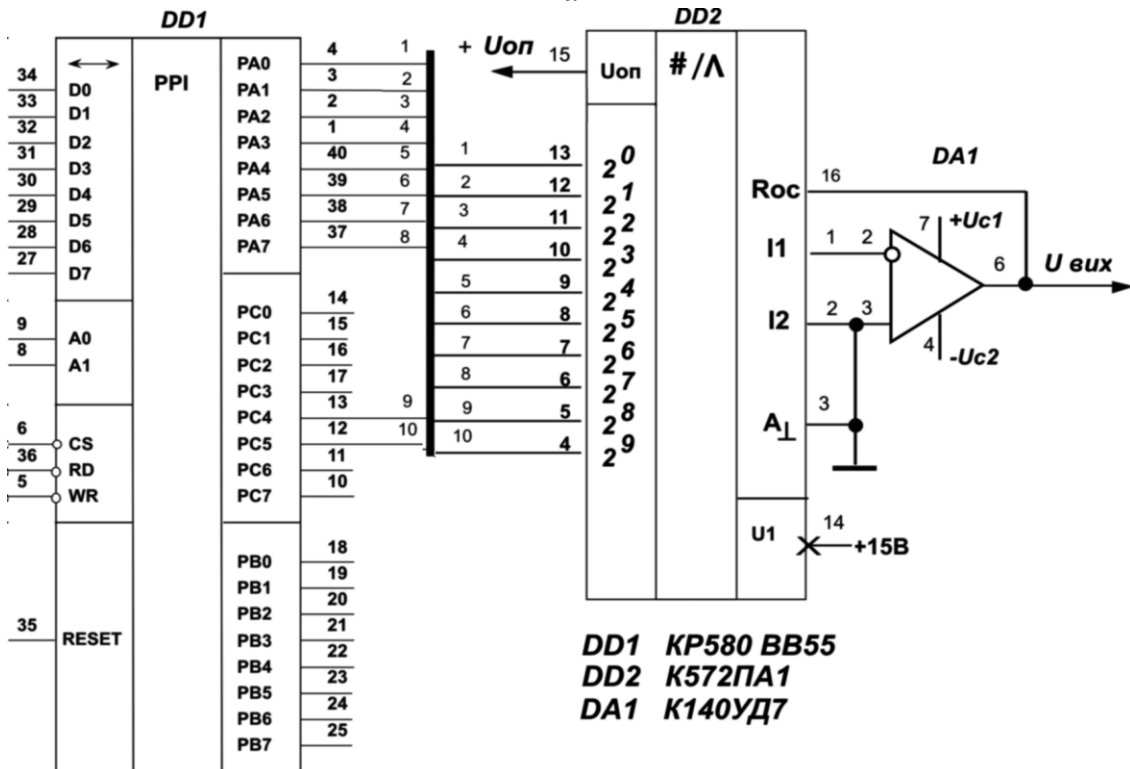
Буферний регістр призначений для зберігання вхідного коду, що

надходить від пристрою виводу. У випадку, коли функції ПВВ виконує, наприклад, програмований паралельний інтерфейс (КР580ВВ55), що має здатність зберігати інформацію на своїх виводах до повторного її виведення, 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 КР580ВВ55.

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	MN/ $\overline{\text{MX}}$		A/D6	10
30	HLDA (RQ/GT1)		A/D7	9
31	HOLD (RQ/GT0)		A/D8	8
			A/D9	7
			A/D10	6
			A/D11	5
			A/D12	4
			A/D13	3
			A/D14	2
		A/D15	39	
		A16/S3	38	
		A17/S4	37	
		A18/S5	36	
		A19/S6	35	
		A18/S5	36	
		A19/S6	35	
		$\overline{\text{BHE/S7}}$	34	
		$\overline{\text{WR(LOCK)}}$	29	
		$\overline{\text{RD}}$	32	
		M/ $\overline{\text{I0}}$ (S2)	28	
		DT/R(S1)	27	
		DEN(S0)	26	
		ALE(QS0)	25	
		$\overline{\text{INTA(QS1)}}$	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 наступній 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
35-38	A19/S6, A18/S5, A17/S4, A16/S3	Старші сигнали шини адреси мультиплексовані з сигналами стану
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

1	2	3
32	\overline{R}	Зчитування. Логічний 0 на виході вказує, що в цьому циклі обміну відбувається зчитування
34	$\overline{BHE}/ST7$	Дозвіл передавання старшої половини каналу даних AD8 – AD15 та сигнал стану
33	MN/\overline{MX}	Мінімальний або максимальний режим. Логічний 0 вказує на роботу МП у максимальному режимі роботи, лог. 1 – у мінімальному
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 передавання 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 нуль – обмін з пристроями вводу-виводу, або з підтвердження переривання
29	\overline{WR}	Запис. Логічний нуль вказує, що відбувається запис
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 бітів 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	Запис у порт
0	1	1	Запис у 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 таких пристроїв:

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

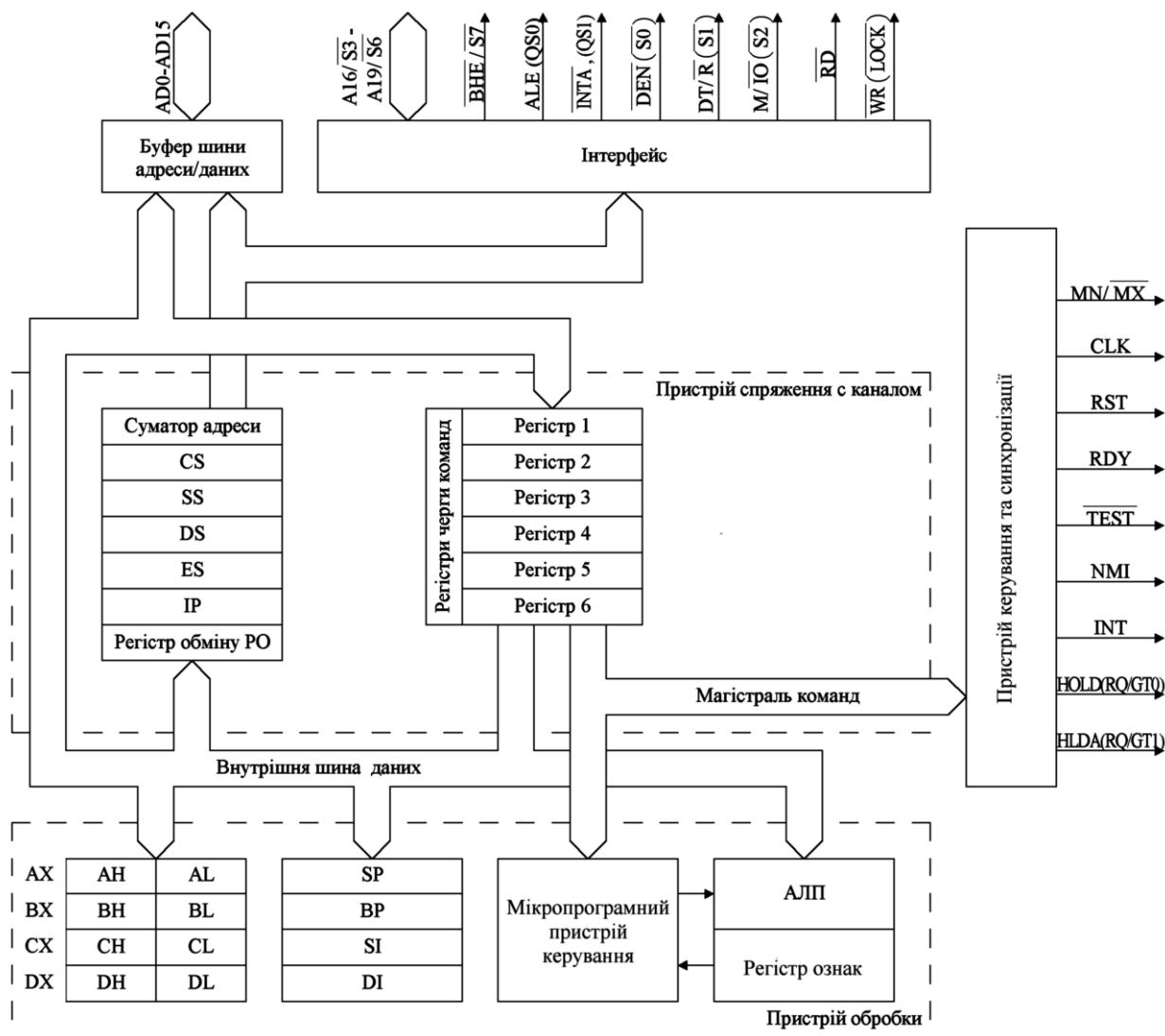


Рис. 28. Структурна схема мікросхеми K1810BM86

Пристрій спряження з каналом складається із:

1. Буфера адреси даних, що призначений для підсилення й 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: сегмента коду (CS), сегмента стеку (SS), сегмента даних (DS), додаткового сегмента (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 бітів. 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. Шістнадцятирозрядного АЛП, де виконуються всі арифметичні та логічні команди.

4. Мікропрограмного пристрою керування, що керує роботою інших блоків МП.

5. Регістра ознак. Регістр ознак вказує стан процесора та ознаки результату виконання останньої арифметичної або логічної операції. Формат регістра ознак зображено на рис. 29. Призначення бітів 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 бітів регістра ознак

Ознака	Значення
AF	Якщо біт дорівнює 1, то відбулося 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 відбулося перенесення або позичання із старшого розряду результату
OF	Якщо біт дорівнює 1, то відбулося 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 К580ВМ80 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 ширшому діапазоні адрес 0-FFFFH (наприклад, команда IN AX, DX).

2.1.6. Адресація пам'яті

Кожна комірка пам'яті, адресована МП К1810ВМ86, має два типи адрес: фізичну, що однозначно визначає її положення в 1Мбайті пам'яті, та логічну. Логічна адреса містить 32 біти й складається із 2 частин – сегмента та зміщення, які є 16-бітними числами. Зазвичай, при програмуванні МП корисуються логічною адресою, яка записується у формі SEG:OFFSET, де SEG – сегментна частина адреси, OFFSET – зміщення. Наприклад: В800:0024, 0040:0000, FF00:0000 тощо. Між фізичною та логічною адресами є проста відповідність:

$$\text{PhA} = 16 \cdot \text{SEG} + \text{OFFSET},$$

де PhA – фізична адреса.

Слід зауважити, що у мікропроцесорі К1810ВМ86 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 розміщуватись 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 К1810ВМ86 становить 64 Кбайт.

У будь-який момент часу програма може звертатися 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 реєстрам ВР	S	S	+	S
Елементарні дані, та індексовані дані, крім даних, де зміщення формується за допомогою реєстра ВР	S	+	S	S
Стек	-	-	+	-
Рядок джерела	S	+	S	S
Рядок приймача	-	-	-	+
Таблиця	S	+	S	S
<p>У таблиці прийняті такі позначення: + – сегмент за замовчуванням; S – сегмент, що може використовуватися за наявності префікса перекриття сегмента; - – сегмент неможливо використати</p>				

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 реєстра ІР.
2. При роботі зі стеком зміщення береться із реєстра 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 й базового регістрів

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

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

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

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

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

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

3.7. При непрямій адресації з індексним і базовим регістрами та зміщенням зміщення даних визначається сумою індексного та базового регістрів, 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 бітів за допомогою копіювання в старший байт сьомого біта зміщення, заданого в команді.

3.8. При табличній адресації адреса формується шляхом сумування вмісту регістра AL з регістром 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 регістр SI.

3.10. При адресації рядка, в яку вміщаються дані (рядка-приймача), в якості зміщення обов'язково використовується регістр 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 використовуються 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 $\overline{ВНЕ}$, як наведено в таблиці 16.

Таблиця 16. Вибірка банків пам'яті

$\overline{ВНЕ}$	A0	Банк, що вибирається	Лінії шини даних, по яких передаються дані
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 поділяються 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 відбулося. Цей номер носить назву вектора переривання.

При немаскованому перериванні запит надходить по входу NMI. Опрацювання переривання не залежить від стана біта IF регістра ознак. Немасковані переривання використовуються для повідомлення процесора в „загрожуючих” режимах (можливість від’єднання живлення, помилка пам’яті й т.д.). Для переривань NMI немає підтвердження, а МП реагує на перехід у 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 поділяють на:

1. Переривання за відлагоджувальними точками.
2. Переривання за переповненням.
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 дорівнює 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 КР1810ВМ86

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 відлагоджувальною точкою 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 бітів) та сегмента (старші 16 бітів) 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 сегментом адреси початок 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 скористатися 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 уміст лічильника команд, уміст регістра ознак та уміст 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 KP580BM80 робота мікропроцесора K1810BM86 здійснюється по тактах, проте в МП K1810BM86 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, за якою відбувається обмін. Потім (з деякою затримкою відносно фронту сигналу CLK) на лінії ALE встановлюється лог. 0. Перехід сигналу ALE в стан логічного нуля вказує, що адресу, видану на шину, необхідно зафіксувати. За спадом сигналу CLK лінії AD0-AD15

переводяться на прийом даних, а на лінії A16/S4 – A19/S6 та лінії $\overline{\text{BHE}}/\text{S7}$ видається слово стана.

Після цього на лінії $\overline{\text{R}}$ видається лог. 0, що вказує на прийом даних, а по фронту сигналу CLK на лінії $\overline{\text{DEN}}$ встановлюється лог. 0, котрий дозволяє роботу вихідних буферів. За спадом 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, у протилежному випадку – T3.

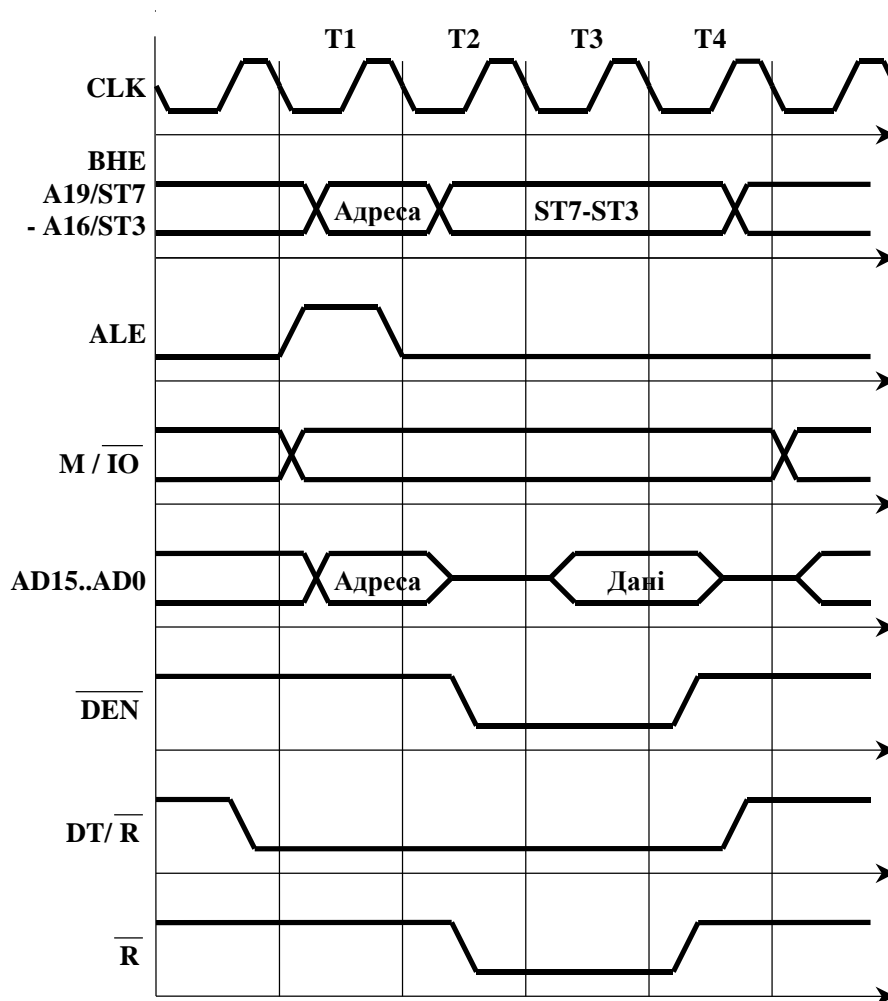


Рис. 30. Часова діаграма роботи МП при читанні в мінімальному режимі роботи

У такті T0 всі лінії зберігають свій стан і за спадом сигналу CLK знову аналізується стан лінії RDY. Якщо лінія в стані лог. 1, то наступний такт – T3, в протилежному випадку повторюється знову такт T0. У такті T4 за спадом

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 запису у пам'ять або у порти введення-виведення в мінімальному режимі роботи

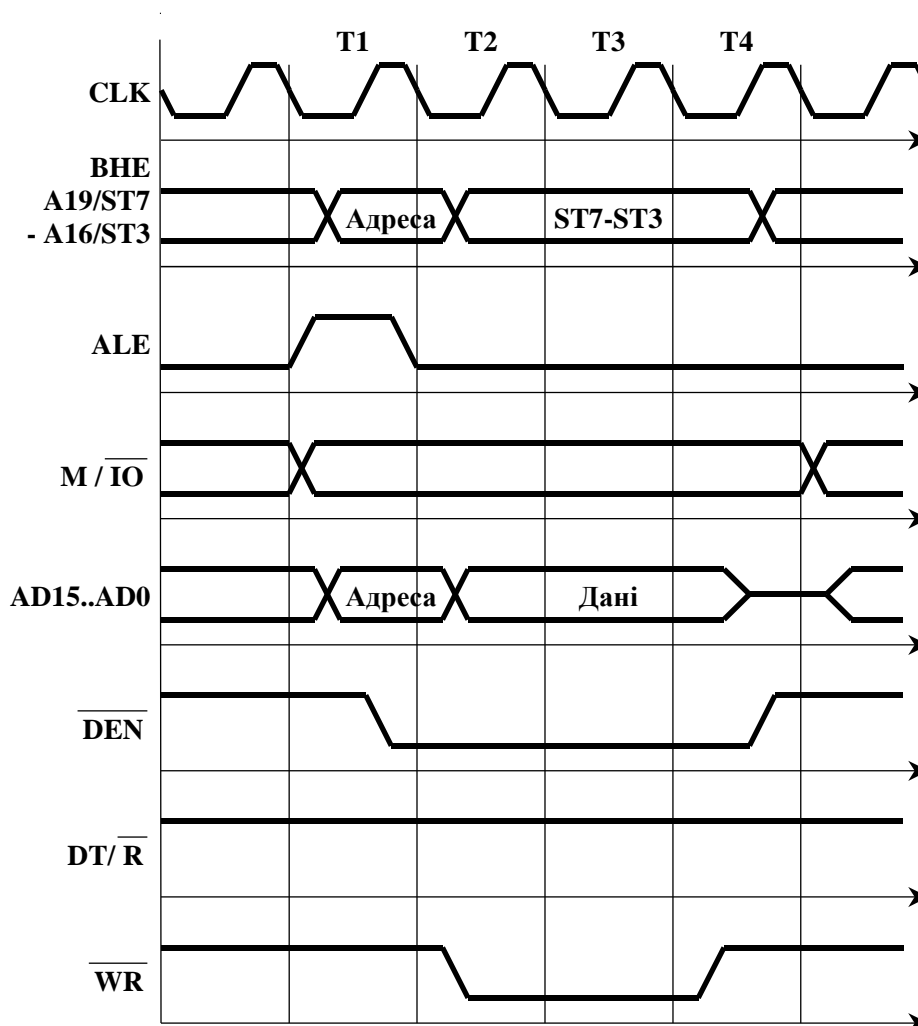


Рис. 31. Часова діаграма роботи МП при запису в мінімальному режимі

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

Подібно до часової діаграми читання, у кінці машинного такту, що передує початку машинного циклу, який розглядається, на лінію DT/ \overline{R}

видається сигнал, що вказує на наявність передавання даних у наступному циклі (лог. 1). На початку машинного циклу за спадом 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 $\overline{BHE}/S7$ видається адреса, за якою відбувається обмін. Потім, з деякою затримкою відносно фронту сигналу CLK на лінії ALE, встановлюється лог. 0, а на лінію \overline{DEN} видається лог. 0, що дозволяє роботу вихідних буферів. Перехід сигналу ALE в стан лог. 0 вказує, що адресу, видану на шину, необхідно зафіксувати. За спадом сигналу CLK на лінії AD0-AD15 видаються дані, що записуються, а на лінії A16/S4 – A19/S6 та лінії $\overline{BHE}/S7$ – слово стана. Після цього на лінії \overline{WR} видається лог. 0, вказуючи на запис даних. За спадом 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 за спадом сигналу CLK знову аналізується стан лінії RDY. Такти T0 повторюються, поки лінія не перейде в стан лог. 1. У такті T3 сигнали залишаються незмінними. Наступним за T3 іде такт T4. Протягом такту T4 мікропроцесор знімає сигнали керування в такому порядку: на початку такту знімається сигнал \overline{WR} ; по фронту сигналу CLK лінія DT/ \overline{R} переходить у стан, що відповідає наступному обміну, з шини даних знімаються дані, що записувались, а лінія \overline{DEN} переводиться в пасивний стан. На цьому обмін у режимі запису 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} .

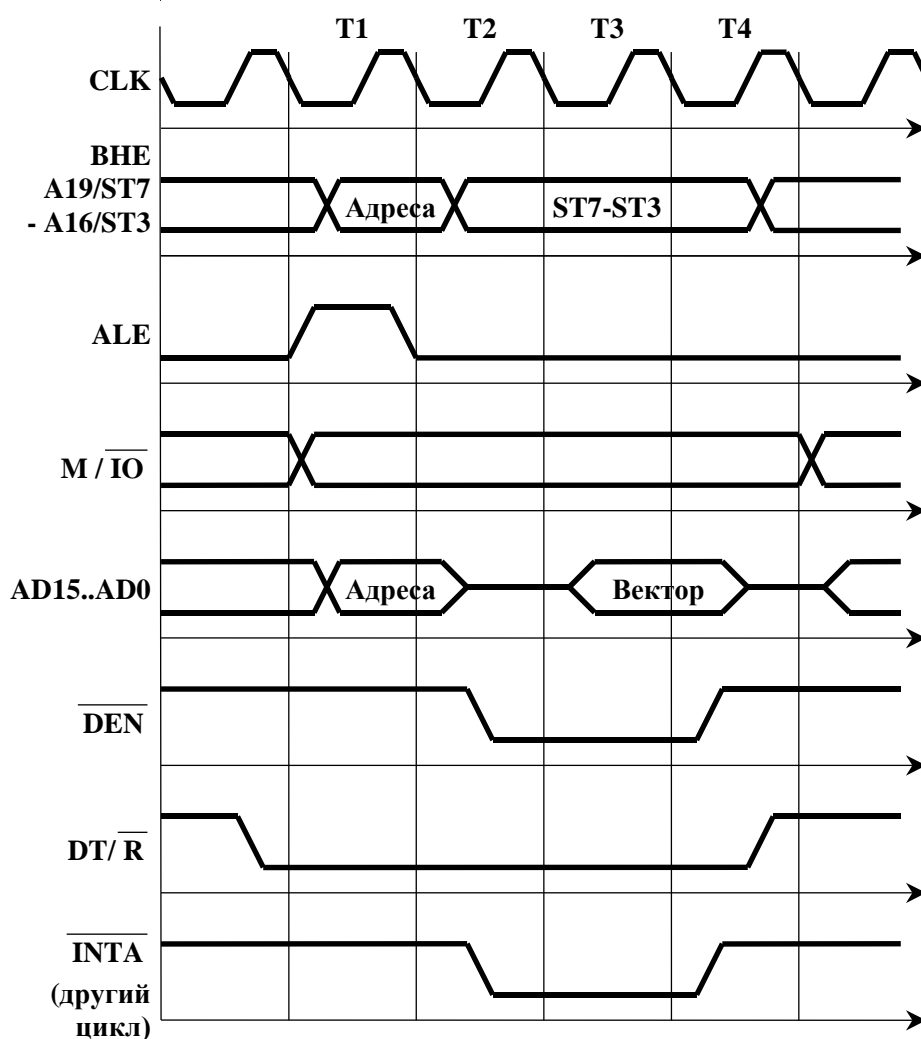


Рис. 32. Часова діаграма роботи МП при підтвердженні переривання в мінімальному режимі роботи

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.

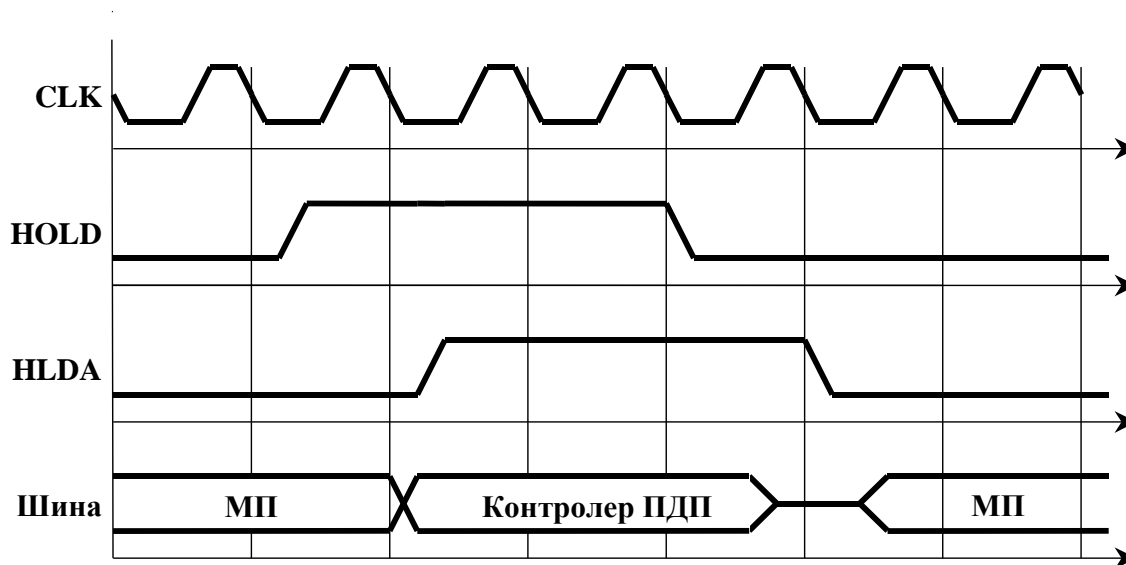


Рис. 33. Часова діаграма роботи МП при прямому доступі до пам'яті в мінімальному режимі роботи

У режим захоплення шин у мінімальному режимі роботи МП входить за сигналом $HOLD = 1$ 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 у 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

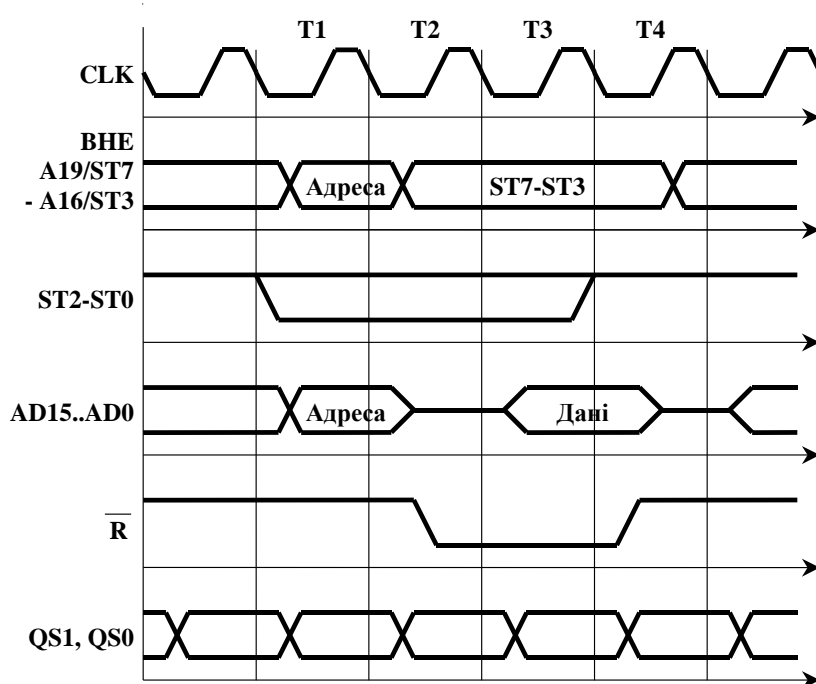


Рис. 34. Часова діаграма роботи МП при читанні в максимальному режимі роботи

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

У кінці машинного такту, що 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 за спадом CLK з певною затримкою відносно нього, на шині AD0-AD15, лініях A16/S4 – A19/S6 та лінії $\overline{BHE}/S7$ видається адреса, за якою відбувається обмін. На початку другого машинного такту лінії AD0 – AD15 переводяться на

приймання даних, а на лінії A16/S4 – A19/S6 та лінії $\overline{\text{BHE}}/\text{S7}$ видається слово стана. Після цього на лінії $\overline{\text{RD}}$ видається лог. 0, який вказує на приймання даних. За спадом 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 T0, в протилежному випадку – T3. У такті T0 усі лінії зберігають свій стан і в кінці такту за спадом сигналу CLK знову аналізується стан лінії RDY. Якщо лінія в стані лог. 1, то наступний такт T3, у протилежному випадку знову повторюється такт T0. На початку такту T3 за спадом сигналу CLK лінії $\overline{\text{S0}} - \overline{\text{S2}}$ 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 $\overline{\text{RD}}$ – зчитування інформації завершено.

2.1.10.2. Часова діаграма запису у пам'ять чи порти вводу-виводу в 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 $\overline{\text{S0}} - \overline{\text{S2}}$ видається код стана, що відповідає обміну, який відбудеться. На початку машинного циклу по спаду CLK, з певною затримкою відносно нього, на шині AD0-AD15, лініях A16/S4 – A19/S6 та лінії $\overline{\text{BHE}}/\text{S7}$ – адреса, за якою відбувається обмін. На початку другого машинного такту на лінії AD0-AD15 видаються дані, що записуються, а на лінії A16/S4 – A19/S6 та лінії $\overline{\text{BHE}}/\text{S7}$ – слово стана. На початку такту T3 за спадом сигналу CLK лінії $\overline{\text{S0}} - \overline{\text{S2}}$ переводяться в пасивний стан. А в 4 машинному такті по фронту сигналу CLK дані, видані на

лінії AD0-AD15, знімаються – запис інформації завершено. Між тактами T2 та T3 можуть бути додані такти очікування (TO), якщо лінія RDY знаходиться в стані лог. 0. Аналіз лінії RDY здійснюється на початку кожного такту TO або T3.

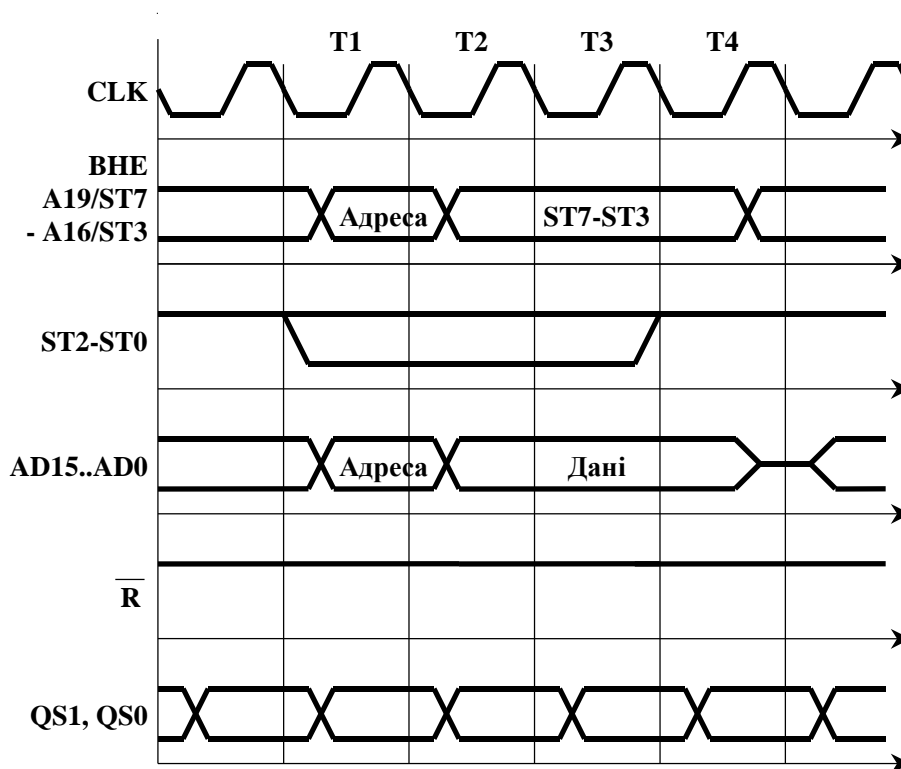


Рис. 35. Часова діаграма роботи МП при записі в максимальному режимі

2.1.11. Початкова ініціалізація

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

Регістр ознак	F	F002h
Регістр адреси 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, запису в пам'ять, завантаження в акумулятор елементів рядків даних. Команди опрацювання рядків дозволяють обробляти рядки 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.

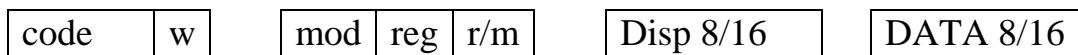
5.4. Команди переривань.

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 байта або слова. 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 у 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 за командою 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 виконується МП. Команда ESC разом із WAIT використовується 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 BHE відсутній, замість нього на лінію видається сигнал стана S0;

17	NMI	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	$\overline{\text{MN/MX}}$		A9	7
			A10	6
30	HLDA		A11	5
	(RQ/GT1)		A12	4
31	HLD		A13	3
	(RQ/GT0)		A14	2
			A15	39
			A16/S3	38
			A17/S4	37
			A18/S5	36
			A19/S6	35
			S0/S7	34
			$\overline{\text{VR(LOCK)}}$	29
			RD	32
			M/D(S2)	28
			DT/R(S1)	27
			DEN(S0)	26
			ALE(QS0)	25
			INTA(GS1)	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,	Старші сигнали шини адреси мільтиплексовані з

	A17/S4, A16/S3	сигналами слова стана
40	+5 В	Напруга живлення
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

Закінчення таблиці 18

1	2	3
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}	Запис. Логічний нуль вказує, що відбувається запис
30	HLDA	Підтвердження захоплення каналу. Логічна 1 вказує, що зовнішній пристрій, який 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 й дозволу доступу до каналу

Стан бітів 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 степеня;
- 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, 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
1,20	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

34	$\overline{BHE} / S7$	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 В	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 й 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 – байта), але ігнорує зчитані дані. Дані та адреса перехоплюються співпроцесором, що фіксує їх у внутрішньому регістрі. Якщо команда вимагає зчитування більше 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 запис даних, то зчитані в першому машинному циклі дані ігноруються і за тією ж адресою співпроцесор, отримавши керування шиною, здійснює запис 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 сприйняти 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 виконують основний процесор та співпроцесор.

2.4. Мікропроцесорний комплект К1810

2.4.1. Структура мікропроцесорного комплекту

У мікропроцесорному комплекті К1810 виділяють мікросхеми таких типів:

1. Мікропроцесори та співпроцесори:

1.1. К1810ВМ86 – 16-розрядний мікропроцесор.

1.2. К1810ВМ88 – 16-розрядний мікропроцесор з 8-розрядною шиною даних.

1.3. К1810ВМ87 – арифметичний співпроцесор.

1.4. К1810ВМ89 – співпроцесор вводу-виводу.

2. Мікросхеми загального призначення (системоутворюючі):

2.1. Генератор тактових імпульсів К1810ГФ84.

2.2. Системний контролер К1810ВГ88.

2.3. Арбітр системної шини К1810ВБ89.

2.4. Регістр з інверсним входом К1810ИР83.

2.5. Регістр з прямим входом К1810ИР82.

2.6. Буферний формувач без інверсії К1810ВА86.

2.7. Буферний формувач з інверсією К1810ВА86.

3. Універсальні інтерфейсні ВІС:

3.1. К1810ВТ37 – контролер ПДП.

3.2. К1810ВН39 – контролер переривань.

3.3. К1810ВІ54 – лічильник-таймер.

4. Мікросхеми-контролери пристроїв та інтерфейсів:

4.1. K1810BG72 – контролер накопичувача на гнучких дисках.

2.4.2. Генератор тактових сигналів K1810ГФ24

Генератор тактових імпульсів (ГТІ) K1810ГФ84 призначений для керування ЦП K1810BM86 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.

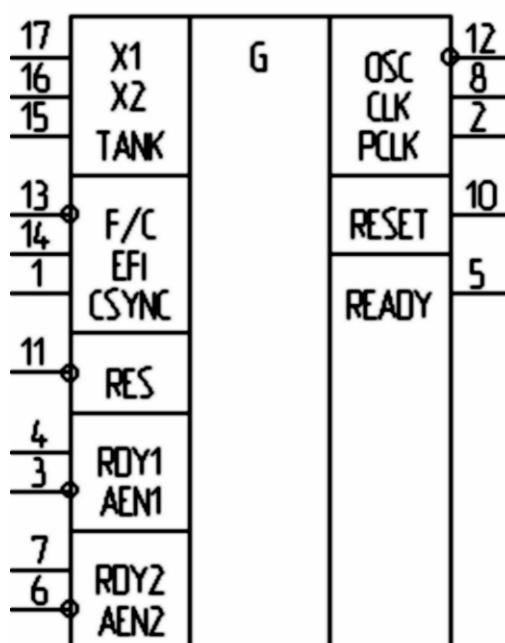


Рис. 37. Умовне позначення мікросхеми K1810ГФ24

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 K1810BM86; PCLK – тактової частоти для керування периферійними ВІС; OSC – сигнал тактової частоти задаючого генератора, необхідні для керування та синхронізації пристроїв, що входять у МП систему.

Частоти сигналів синхронізації пов'язані співвідношенням

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

у режимі внутрішнього генератора та

$$FEFI = 3 FCLK = 6 FPCLK$$

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 у стані лог. 1, то сигнал синхронізації сприймається по входу EFI.

Схема формування скиду RST має на вході тригер Шмітта, а на виході – D-тригер, який формує фронт сигналу RST за спадом 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 за спадом сигналу CLK та формується за законом

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

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	$\overline{AEN1}$	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	$\overline{AEN2}$	Вхід	Адреса готовності 2
8	CLK	Вихід	Тактовий сигнал МП
9	GND	–	Загальний
10	RST	Вихід	Скид МП (низьким рівнем)
11	\overline{RES}	Вхід	Вхід сигналу скиду
12	OSC	Вихід	Вихід мультівібратора
13	F/C	Вхід	Вибір джерела частоти
14	EFI	Вхід	Зовнішня частота

Таблиця 20. Призначення виводів К1810ГФ24

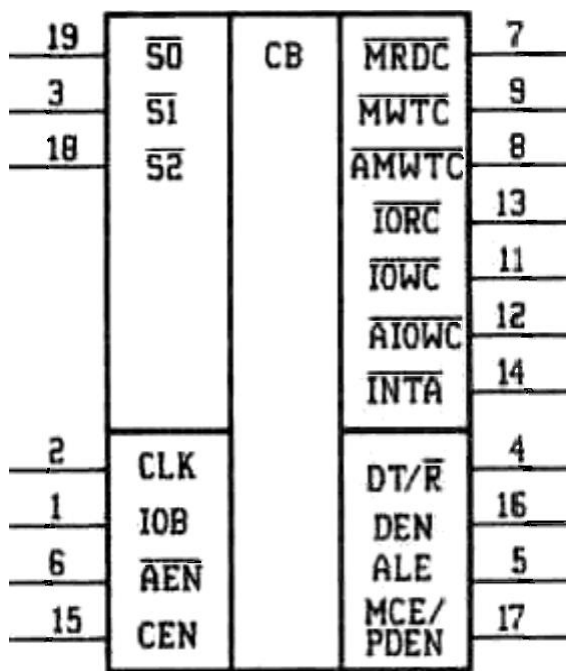
Вивід	Позначення	Тип	Призначення виводу
15	TANK	–	Вивід під'єднання зовнішнього LC резонатора
16	X1	–	Виводи під'єднання зовнішнього кварцового резонатора
17	X2	–	
18	Ucc	–	Напруга живлення

2.4.3. Контролер системної шини К1810ВГ88

Контролер системної шини К1810ВГ88 призначений для роботи в складі мікропроцесорної системи на базі МП К1810ВМ86. 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.



Рис. 38. Структура контролера системної шини К1810ВГ88



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.

Рис. 39. Умовне графічне зображення мікросхеми наведено в таблиці 21.

К1810ВГ88

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

Продовження таблиці 21

1	2	3
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 запису в пам'ять. Логічний 0 вказує на запису в пам'ять. Сигнал встановлюється після видавання даних, що записуються
AMWC	8	Випереджаючий сигнал запису в пам'ять. Логічний 0 вказує на запис у 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 запису в порти вводу-виводу. Лог. 0 вказує на запису
AIOWC	12	Сигнал запису 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
DT/R	4	Сигнал керування роботою шинних формувачів. DT/R = 1 перемикає шинні формувачі на передавання даних. DT/R = 0 перемикає шинні формувачі на приймання
DEN	16	Сигнал дозволу роботи шинних формувачів. Лог. 0 дозволяє їх роботу
ALE	5	Строб адреси. Задній фронт сигналу вказує на справність адреси

Закінчення таблиці 21

1	2	3
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
-----------------------	---------	----------------------------------

Таблиця 22. Декодування коду стану

Код стану S2 S1 S0	Стан МП	Командний сигнал контролера ВГ88
000	Підтвердження переривань	$\overline{\text{INTA}}$
001	Ввід з ПБВ	$\overline{\text{IORC}}$
010	Вивід з ПБВ	$\overline{\text{IOWC}}, \overline{\text{AIOWC}}$
011	Зупинка	-
100	Вибірка команди	$\overline{\text{MRDC}}$
101	Читання з пам'яті	$\overline{\text{MRDC}}$
110	Запис в пам'ять	$\overline{\text{MWTC}}, \overline{\text{AMWC}}$
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 ($\text{IOB} = 0$). У цьому режимі контролер формує командні сигнали та сигнали ALE, DEN, DT/R керування фіксаторами адреси й шинними формувачами. На входах формуються постійні значення $\text{AEN} = 0$ та $\text{SEN} = 1$, 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 SEN. Зазвичай він застосовується у випадках, коли МП має доступ до двох шин: системної та резидентної, причому кожній із них відповідає свій адресний простір. Для формування сигналу SEN у таких випадках використовується дешифратор адреси, що розділяє адресний простір між системною (I-41) та резидентною шиною. Вихід цього дешифратора (прямий та інверсний) служить в якості сигналу SEN.

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 (I -41) тільки по сигналу

AEN від арбітра шини.

Командні сигнали \overline{IORC} , \overline{IOWC} , \overline{AIOWC} та \overline{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{MRDC} або \overline{MWTC} , \overline{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{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 $S_2S_1S_0 = 000$, 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 сигналом ALE записується у фіксатор адреси.

По фронту іншого імпульсу \overline{INTA} ведений контролер, який запитав 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 запису біжучої адреси у фіксатори адреси. Цей сигнал використовується також для фіксації коду стана $S_2 S_1 S_0$ 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 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 мінімальному режимі роботи (рис. 40) МП K1810 BM86 використовує для керування обміном даних по шині власні лінії керування:

- ◇ WR, RD – відповідно сигнали запису 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 запису адреси в адресні регістри адресної шини.

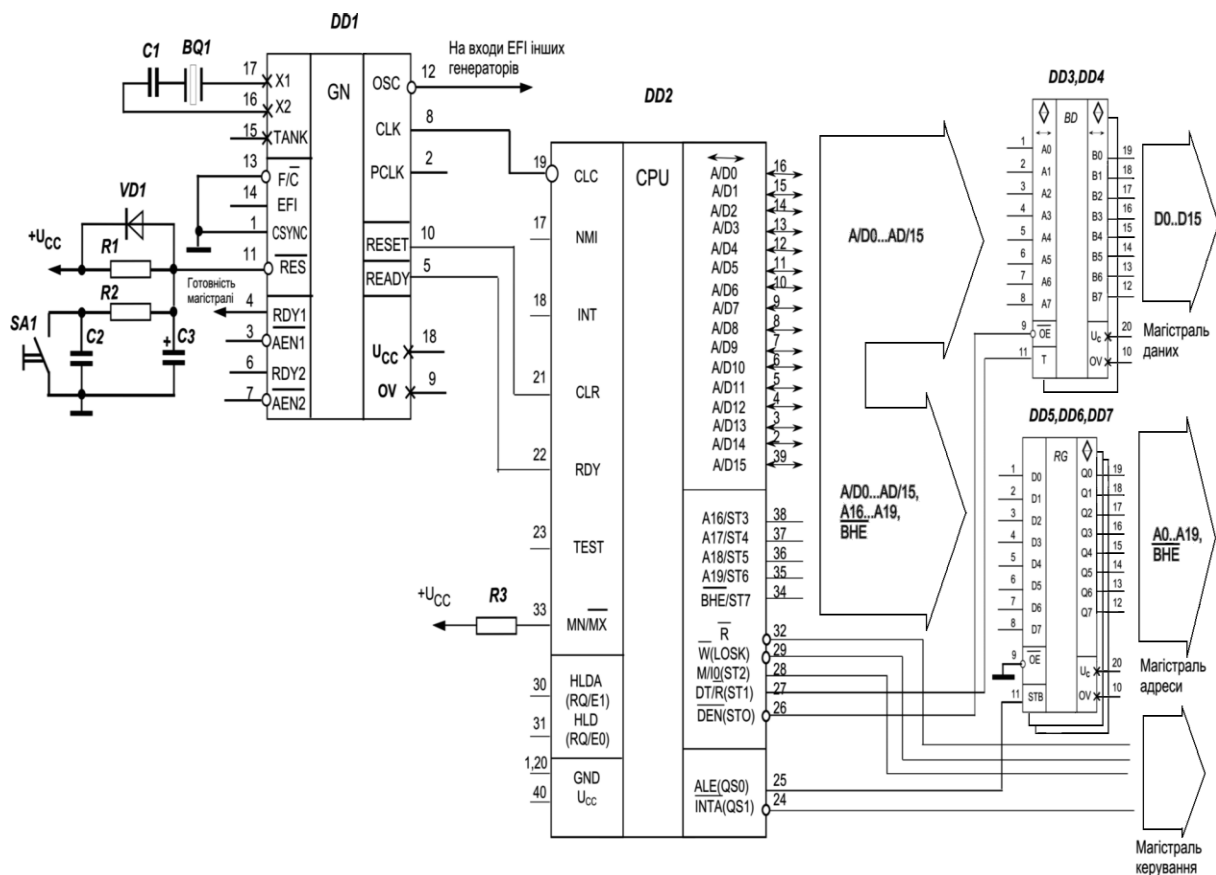


Рис. 40. Структура системної магістралі на базі МП К1810ВМ86 у мінімальному режимі роботи

До складу магістралі в мінімальному режимі роботи МП входять:

- ◇ тактовий генератор КР1810ГФ84 (мікросхема DD1);
- ◇ мікропроцесор КР1810ВМ86 (DD2);
- ◇ шинні формувачі КР580ВА86 (DD3, DD4);
- ◇ адресні регістри КР580ИР82 (DD5-DD7).

2.5.2. Максимальний режим

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 КР1810ВГ88. 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 КР1810ВБ89.

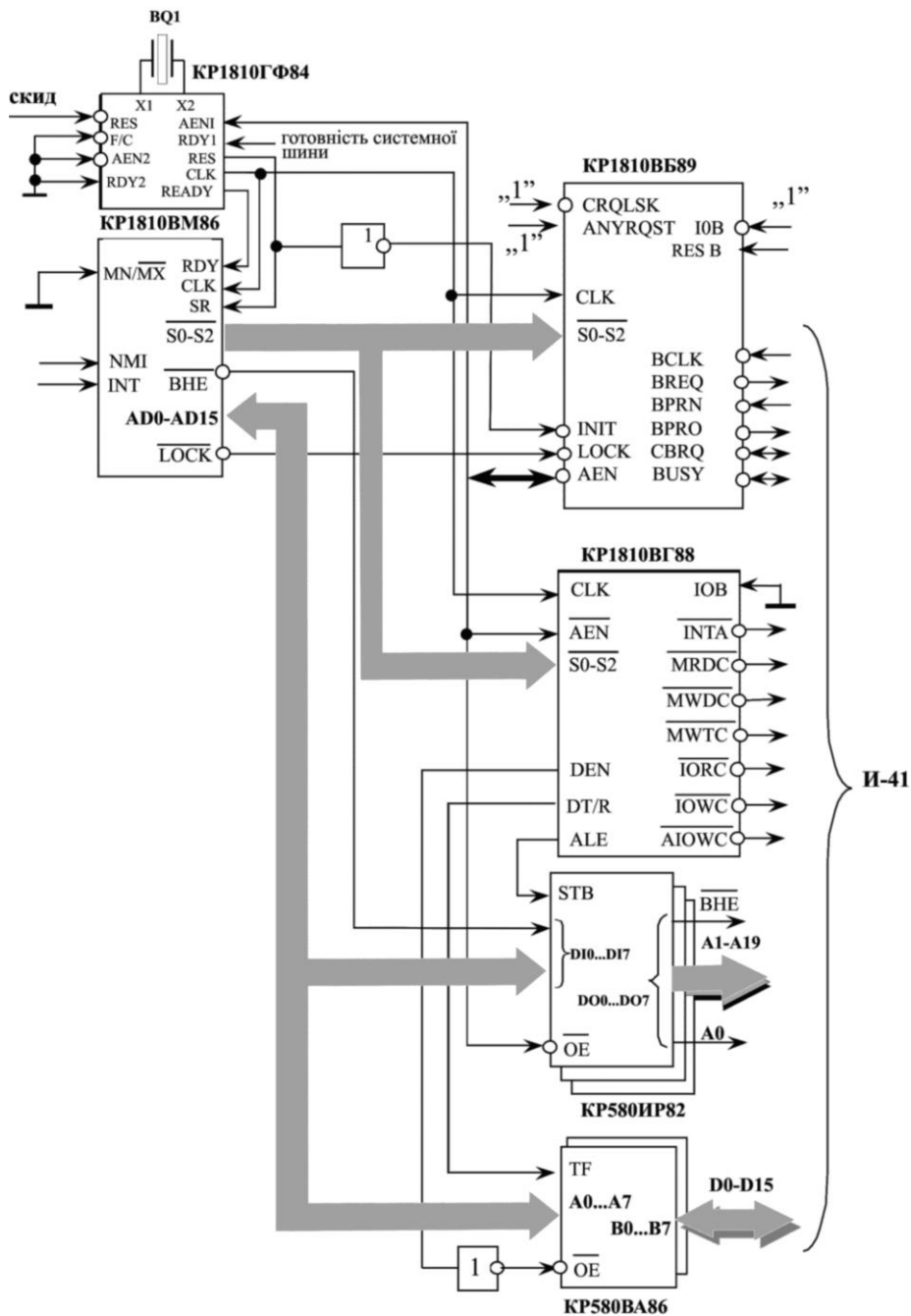


Рис. 41. МП система в максимальному режимі з однією системною магістраллю

МП-система (рис. 41) призначена для роботи з однією системною магістраллю. МП КМ1810ВМ86 працює в максимальному режимі ($MN/MX = 0$) і до його локальної магістралі безпосередньо під'єднані арбітр магістралі КР1810ВБ59, контролер КР1810ВГ88, буферні регістри КР580ИР82 та шинні

формувачі КР580ВА86. До системної магістралі під'єднані розділені ресурси (пам'ять та зовнішні пристрої), а також ін. 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 КР580ВМ80А.

2.5.3. Арбітр магістралі КР1810ВБ89

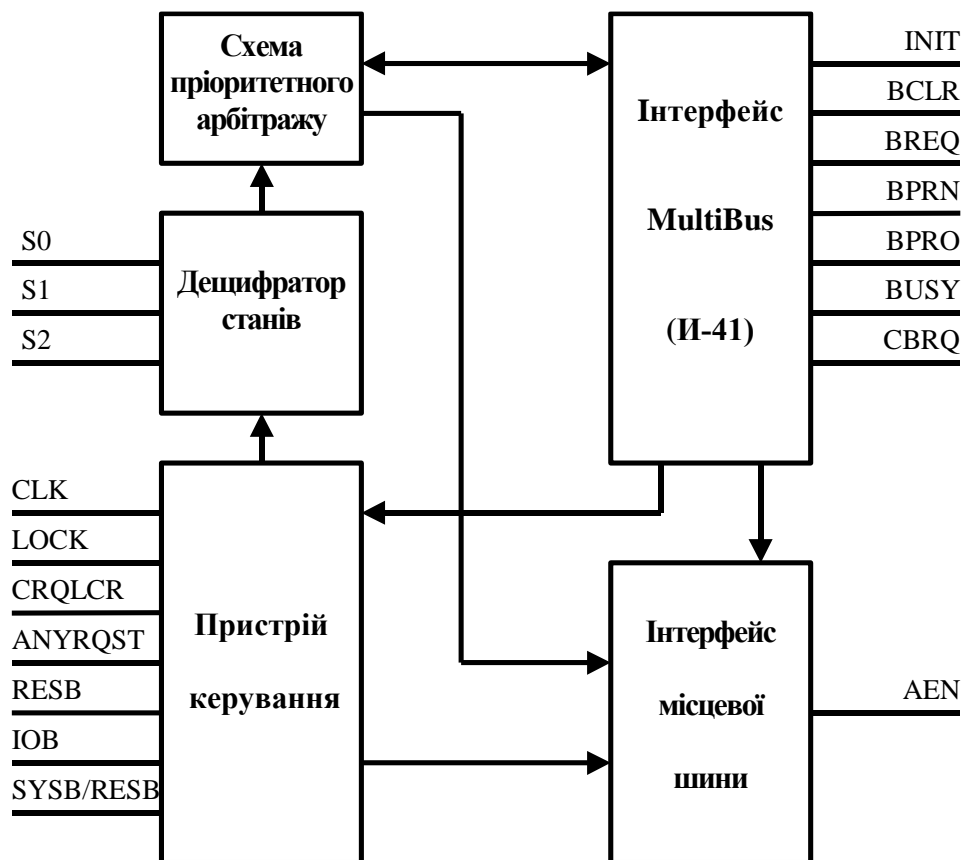


Рис. 42. Структурна схема АШ ВБ89

Арбітр шини забезпечує синхронізацію доступу кількох ведучих пристроїв до системної шини відповідно до визначеного пріоритету доступу. Арбітр шин К1810ВБ89 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 (ВМ86, ВМ87 або ВМ89) запускає схему пріоритетного арбітражу, інтерфейсу И-41 та місцевої шини на виконання дій із захоплення або звільнення системної шини. Схема пріоритетного арбітражу

проводить арбітраж мікропроцесорів, що запитують керування системною шиною, та за тактовим сигналом мікропроцесора, що займає шину, здійснює дії з її звільнення. Інтерфейс И-41(Multibus) здійснює процедуру взаємодії арбітрів шини багатопроцесорної системи та синхронізує дії із захоплення системної шини.

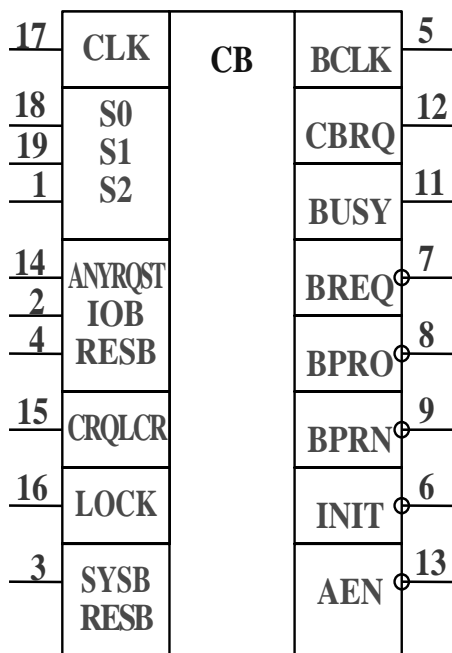


Рис. 43. Умовне графічне позначення АШ К1810Б89

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.

2.5.3.1. 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

1	2	3
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 звільнити 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 шини. 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 шинами. 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 із захоплення та звільнення системної шини. При SYSB/RESB = 1 арбітр запитує системну шину, при SYSB/RESB = 0 він звільняє системну шину іншому арбітру, що запитав керування нею
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

1	2	3
BREQ	7	Сигнал запиту шини. Використовується у схемах паралельного та циклічного дозволу пріоритету для запиту 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 із 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 стана 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 передавання 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 комірka по лінії готовності RDY1 повертає сигнал підтвердження, за яким тактовий генератор KP1810ГФ84 формує сигнал READY. При отриманні цього

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

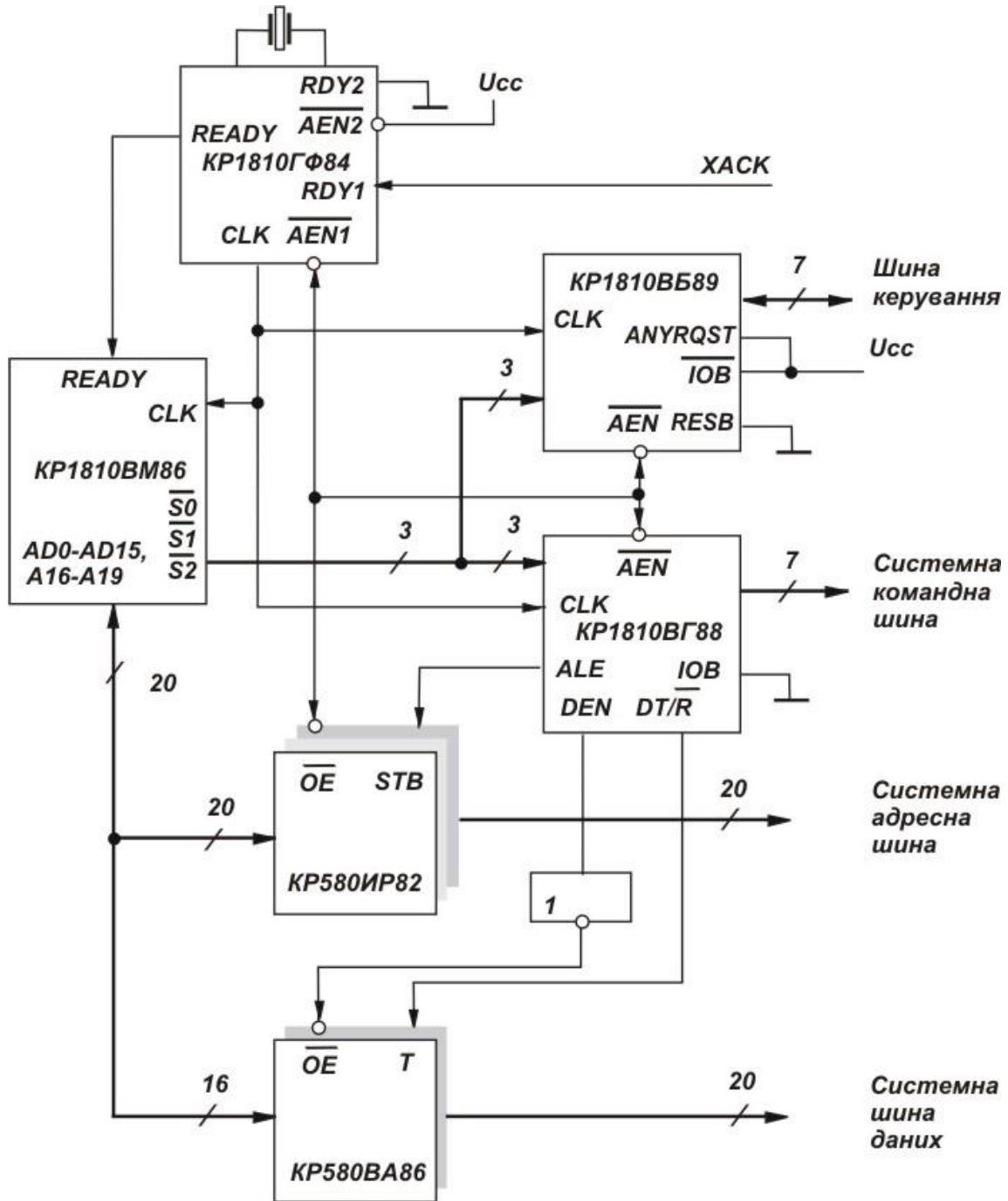


Рис. 44. Арбітр шини в конфігурації з системною шиною

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 АЕН=1 арбітра КР1810ВБ89 забороняє тактовому генератору КР1810ГФ84 видавати сигнал готовності READY в МП.

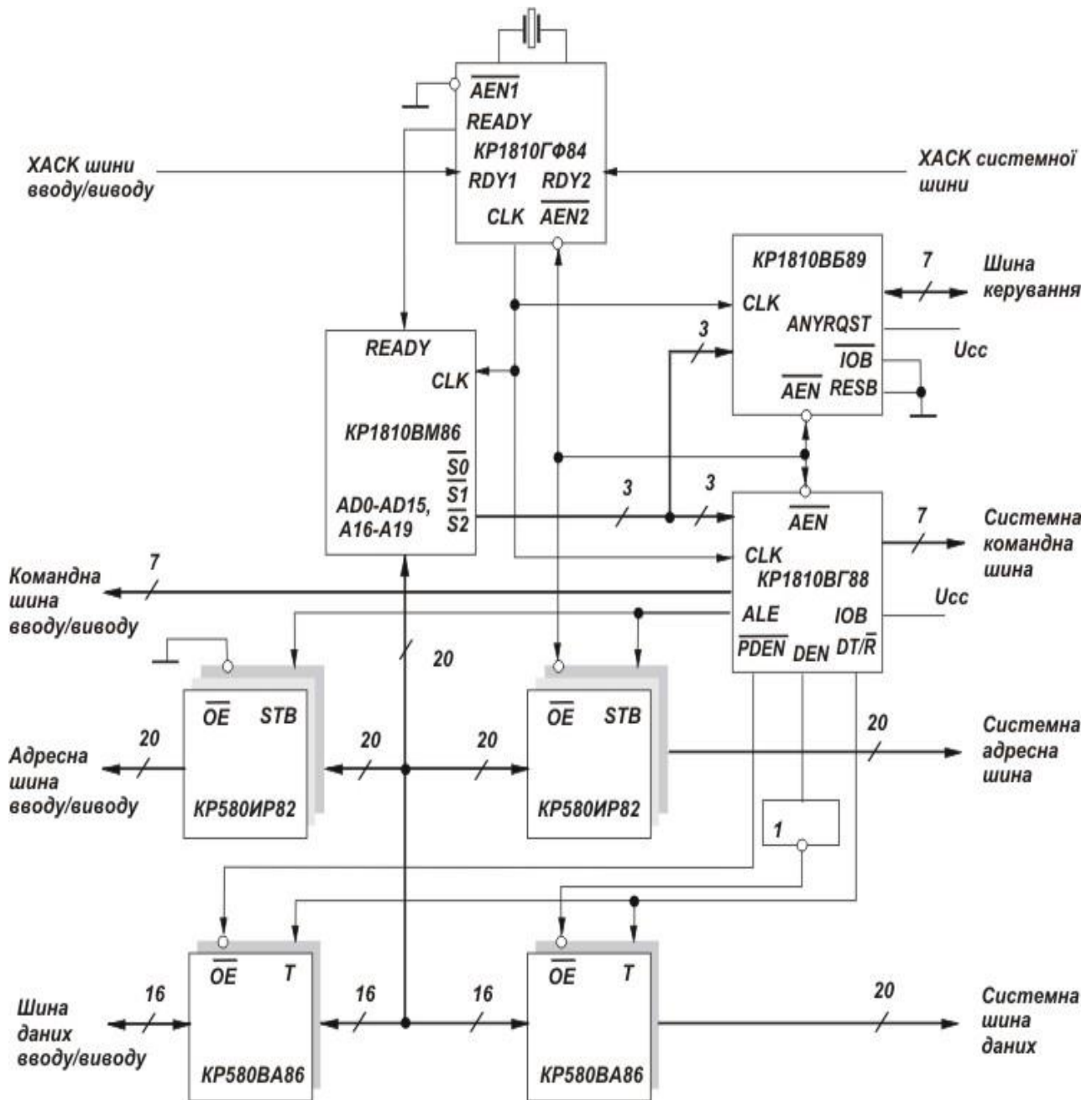


Рис. 45. Арбітр шини в конфігурації з системною та периферійною шинами вводу-виводу

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

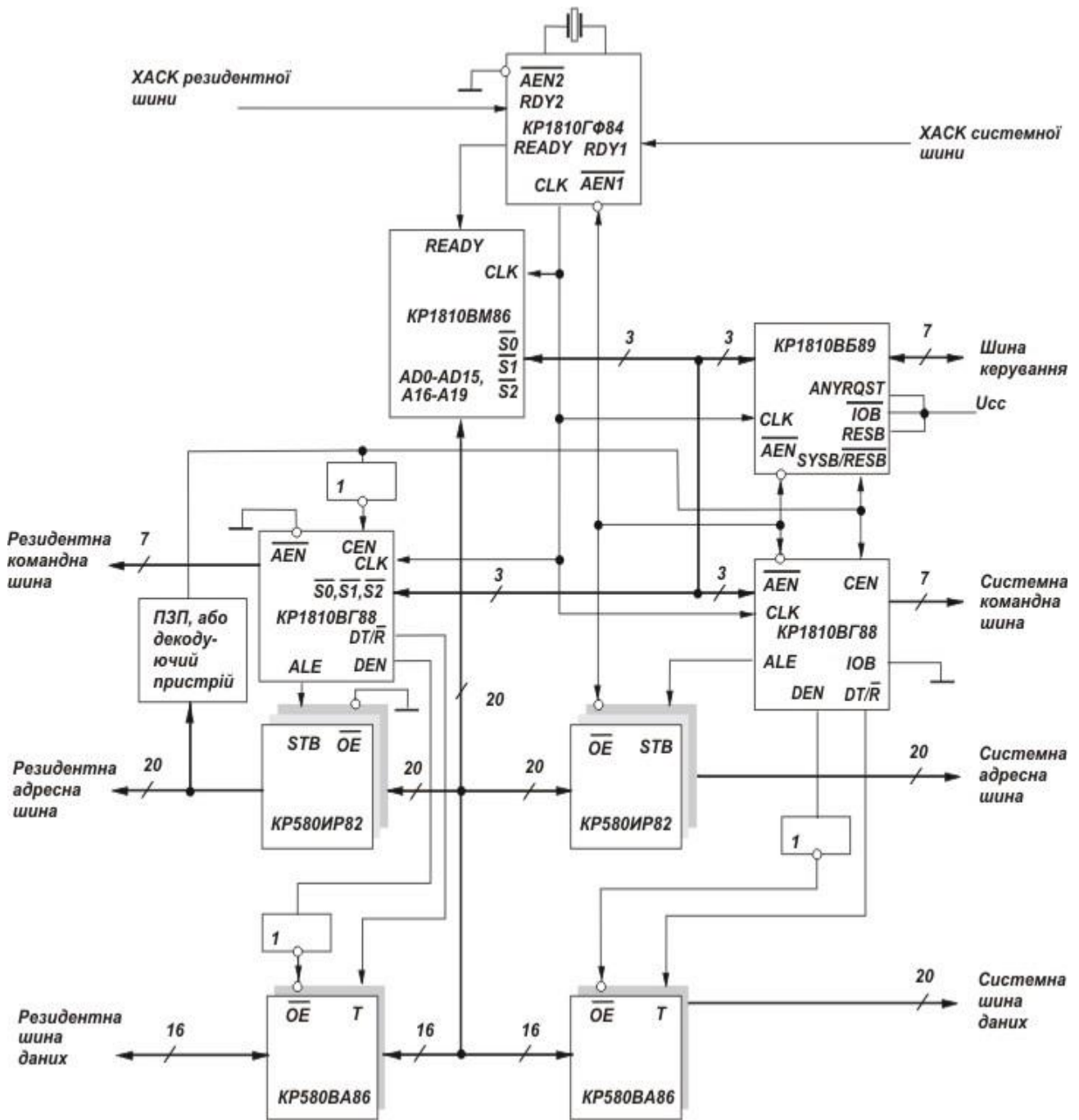


Рис. 46. Арбітр шини в конфігурації з системною та резидентною шинами

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.

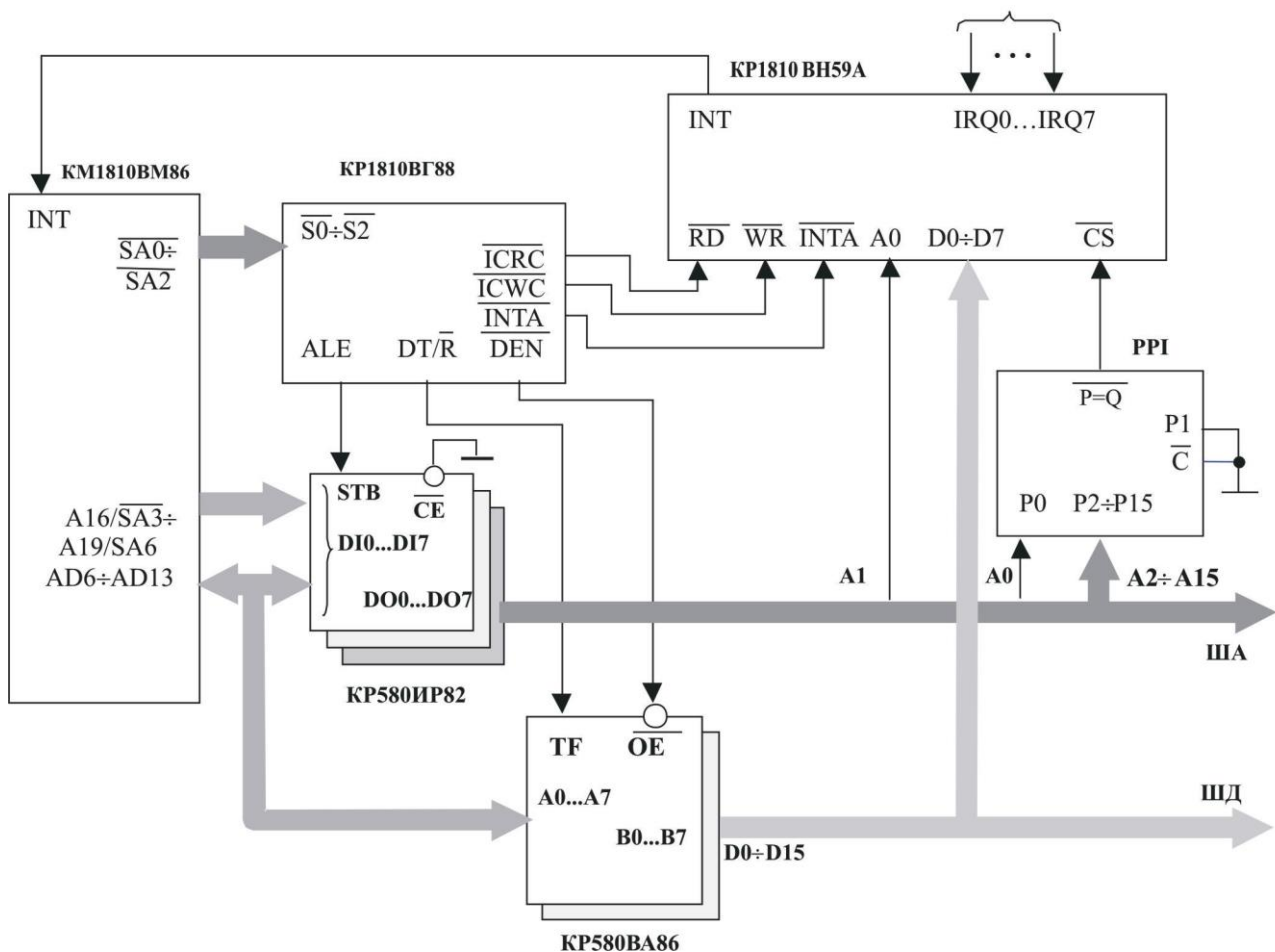


Рис. 47. Під'єднання контролера переривань

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 восьми типів переривань необхідне встановлення кількох 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, запису та вибору мікросхеми.

2.7. Контрольні запитання

1. Розроблення 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 у 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

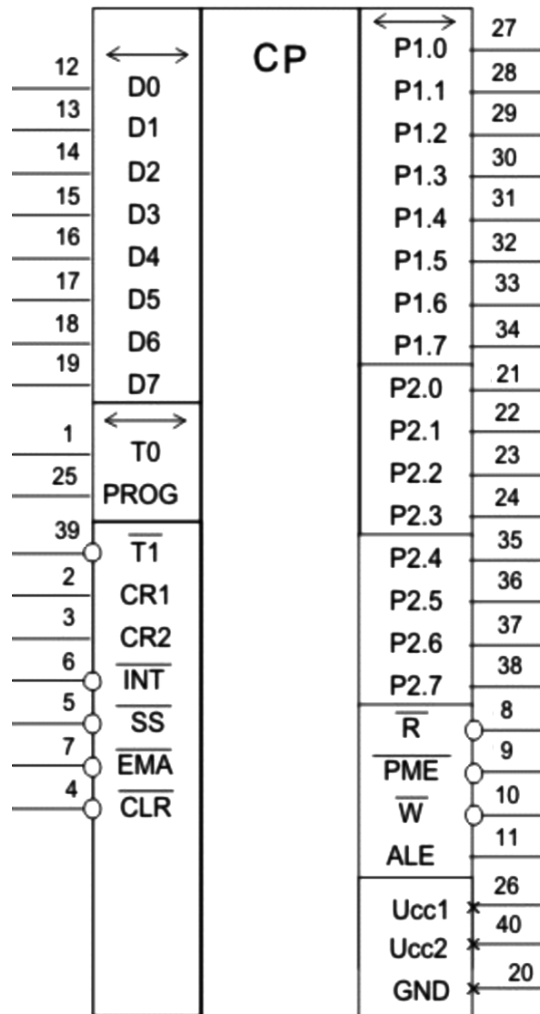


Рис. 48. Умовне позначення КМ1816ВЕ48

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 однокристалної мікроЕОМ КМ1816ВЕ48 зображено на рис. 48, призначення виводів вказано в таблиці 24.

Таблиця 24. Призначення виводів мікросхеми

Позначення	Номер	Призначення
1	2	3
GND	20	Загальний
UCC1	26	Напруга живлення ОЗП у режимі нормальної роботи. У режимі програмування – напруга програмування
UCC2	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 КМ1816ВЕ48 можна подавати сигнал синхронізації від зовнішнього джерела
$\overline{\text{CLR}}$	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 відбуватися при переході сигналу з високого рівня в низький. Сигнал формується в кожному машинному циклі й може використовуватися 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
---	---	---

Закінчення таблиці 24

1	2	3
W	10	Запис. Логічний нуль вказує, що відбувається запис 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 опитуватися командою умовного переходу JN1
EMA	7	Сигнал вимкнення 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 запису 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

зі входами й виходами ТТЛ мікросхем.

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

- ◇ зчитування стану лінії (команди вводу з порту);
- ◇ зчитування стану внутрішнього регістра.

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

KM1816BE48

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

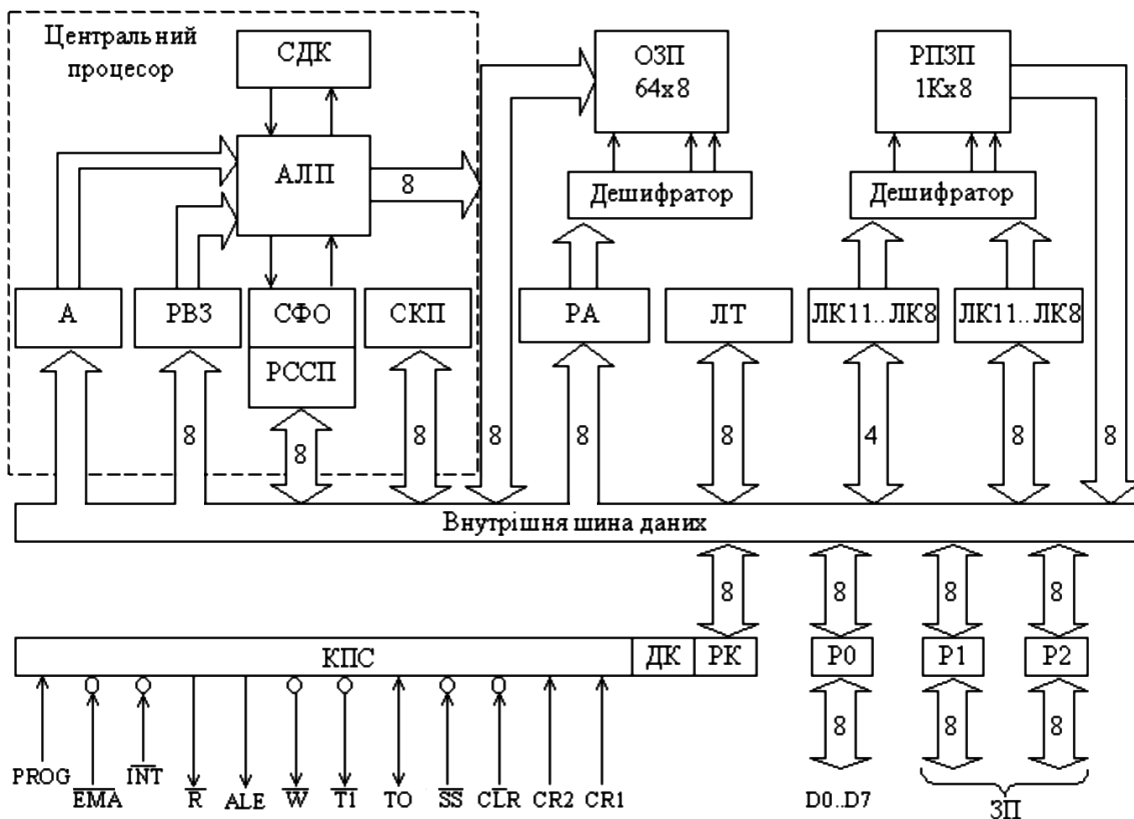


Рис. 49. Структурна схема мікросхеми

Блок "Центральний процесор". Центральний процесор має розрядність 8 бітів 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 стана процесора (РССП) призначений для збереження ознак стану процесора. Формат регістра слова стана зображений на рисунку 50, призначення кожної ознаки наведено в таблиці 25.

Таблиця 25. Значення ознак слова стана

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 – може бути встановлена й 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 1816 – 1F16.
S2, S1, S0	Вказівник стека. Регістр показує адресу біжучої вершини стека. Адреса вершини стека обчислюється за формулою $S2S1S0 \times 2 + 8$

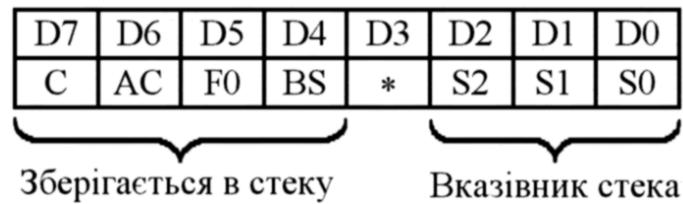


Рис. 50. Формат слова стана програми

Пристрій керування і синхронізації (ПКС), призначений для керування й 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.

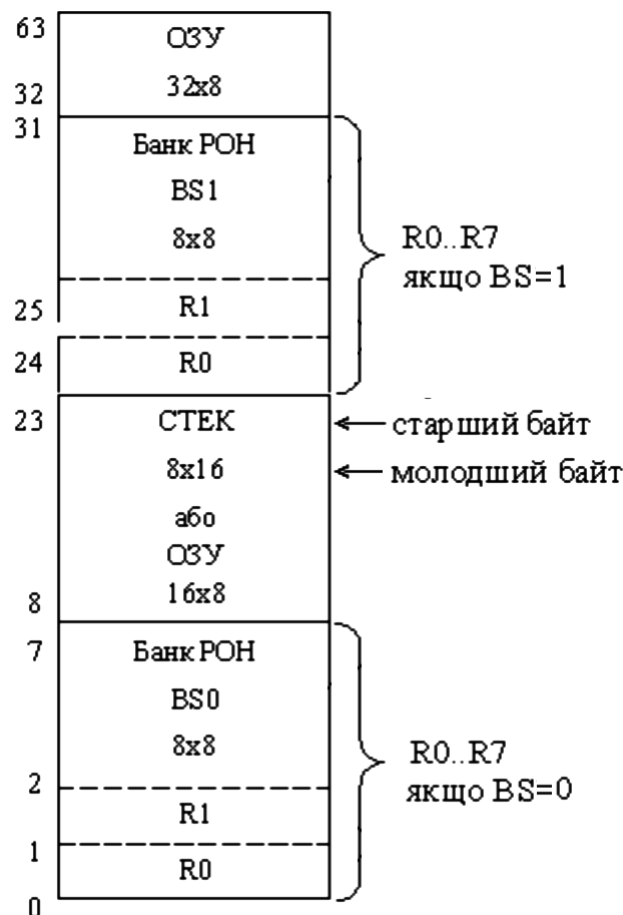


Рис. 51. Структура ОЗП

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 регістрів загального призначення. Усі регістри розбиті на два банки (BS0, BS1) по вісім у кожному. Перемикання банків здійснюється бітом BS регістра слова стана. 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 – постійна 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 до мікропроцесорного комплекту 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

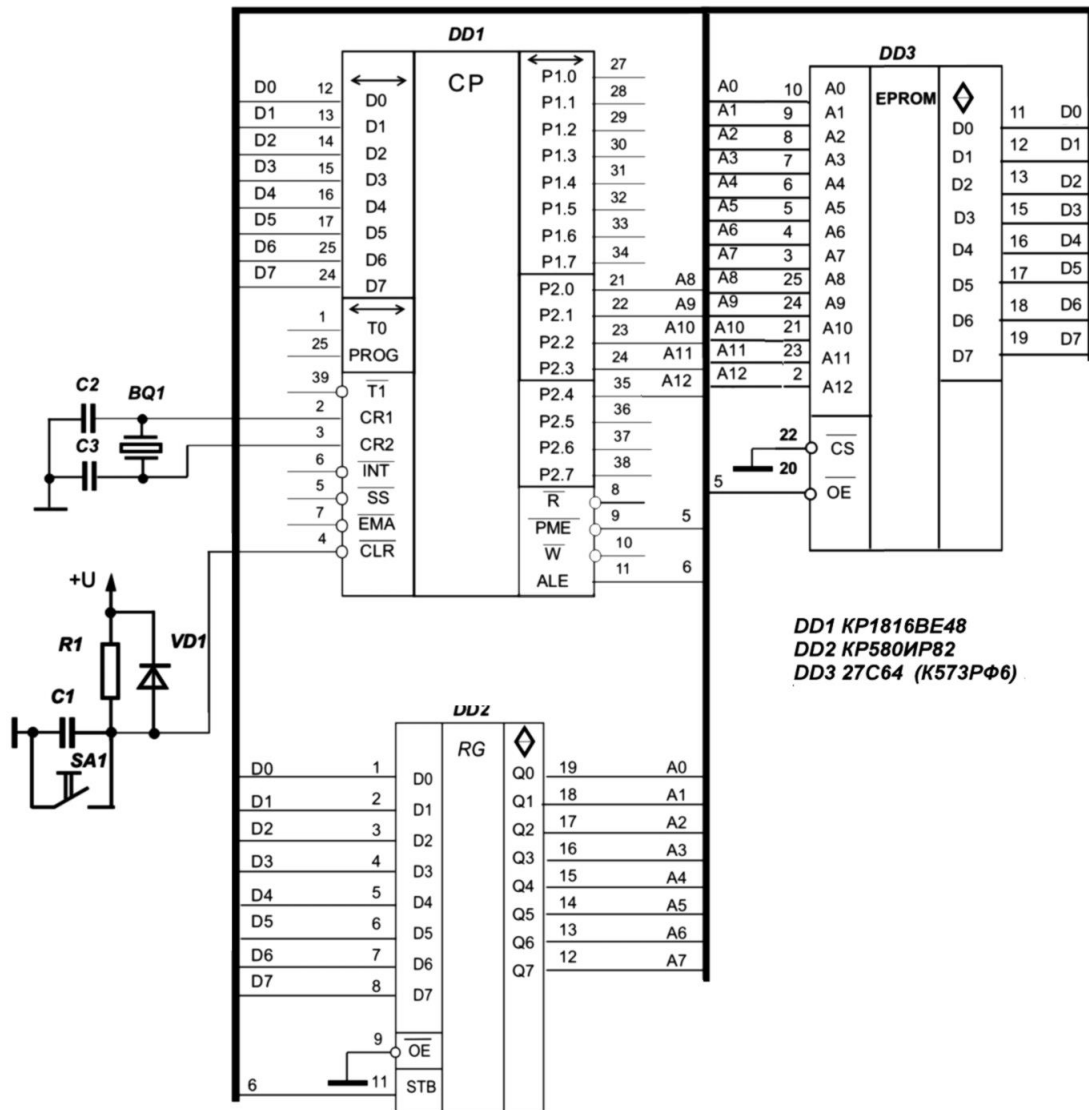


Рис. 52. Під'єднання ПЗП до ОМЕОМ

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);
- ◇ за зрізом сигналу ALE на зовнішньому регістрі фіксується адреса;
- ◇ сигналом PME дозволяється робота зовнішньої пам'яті;
- ◇ за спадом сигналу PME шина BUS переходить у режим вводу.

Комірки пам'яті з адресами, що лежать за межами банку пам'яті 0, будуть доступні після виконання команди перемикавання 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 перемикавання 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 OE заведено сигнал PME.

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 байтів.

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 MCS-48.

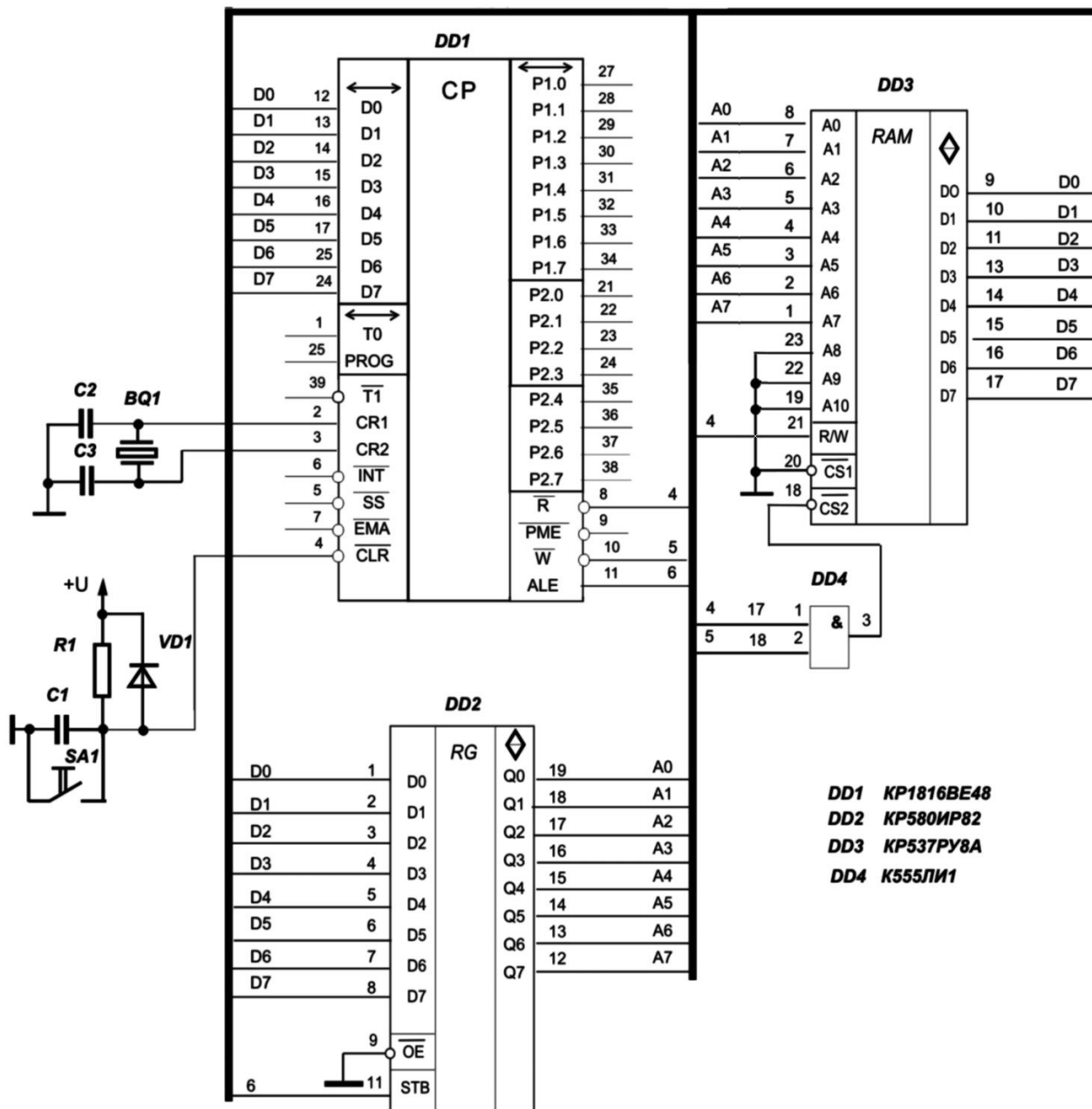


Рис. 53. Зовнішнє ОЗП даних

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).

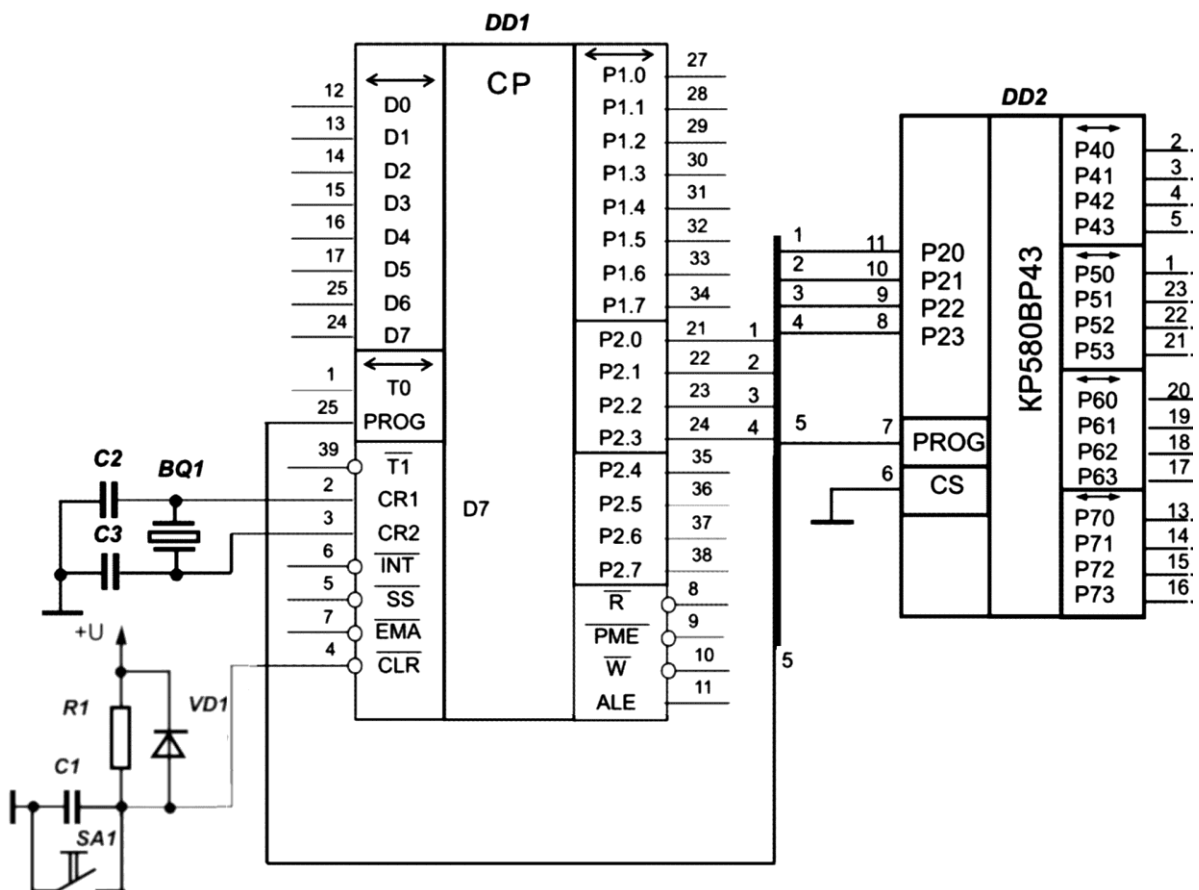


Рис. 54. Використання розширювача вводу-виводу

Розширювач під'єднується до MCS-48 так, як показано рис. 54. Кожен із чотирьох портів РВВ може використовуватися для введення чи виведення інформації незалежно від інших і забезпечувати високу навантажувальну здатність. Для виведення байта даних у порти P4 і P5 розширювача можна

скористатися такою послідовністю команд:

```

MOV D P4, A      ; Вивід A(0...3) в порт 4
SWAP A          ; Обмін тетрад акумулятора
MOV D 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 виведення байта в порт В необхідно виконати дві команди:

```

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

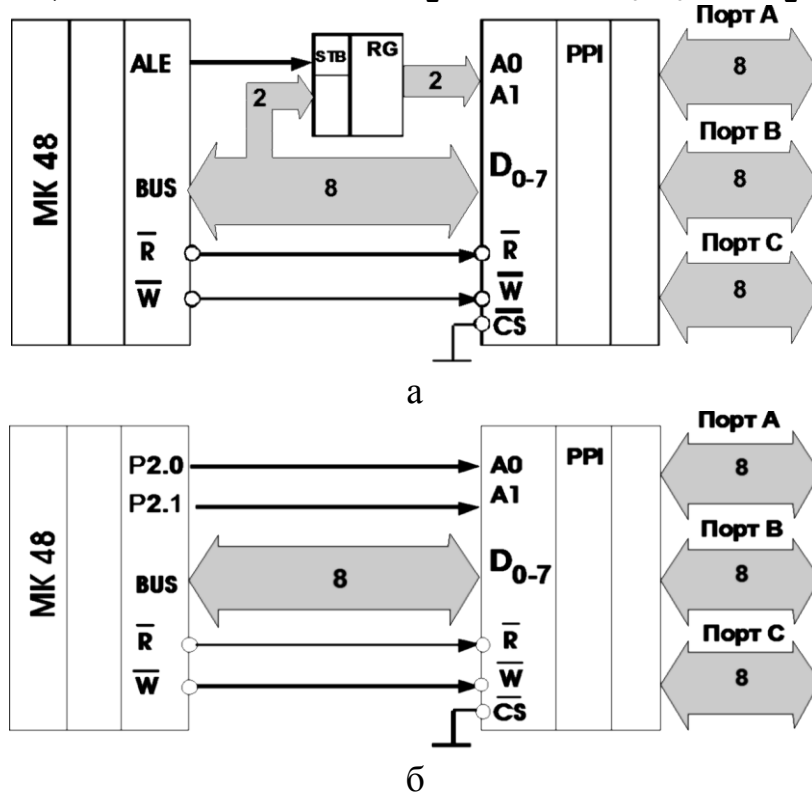


Рис. 55. Використання мікросхем комплексу KP580

Для другого варіанта під'єднання ППА (рис. 55б) вибір порту здійснюється через установку/скидання двох розрядів порту P2. Для виведення 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 кілька 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 введення 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 введення інформації:

- ◇ найпростіші, що складаються з клавіш керування, наприклад, СКИДАННЯ, ПУСК, ЗУПИНКА і т.п.;
- ◇ цифрові, призначені для введення 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 зі зростанням кількості знаків, такі клавіатури в МК-системах застосовуються не часто. Значно ширше використовують дешеві некодуючі (матричні) клавіатури, що є простою матрицею двійкових перемикачів, увімкнених 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 введення коду натисненої кнопки в момент, коли таке звертання необхідне. Процедура введення проводить опитування клавіатури. Викликана з основної програми процедура блокує процес керування об'єктом на час опитування клавіатури.

Процедуру введення 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 зображена 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, який відповідає місцю 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 кількох клавіш.

Процедура введення коду натисненої клавіші складається з послідовності таких дій:

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

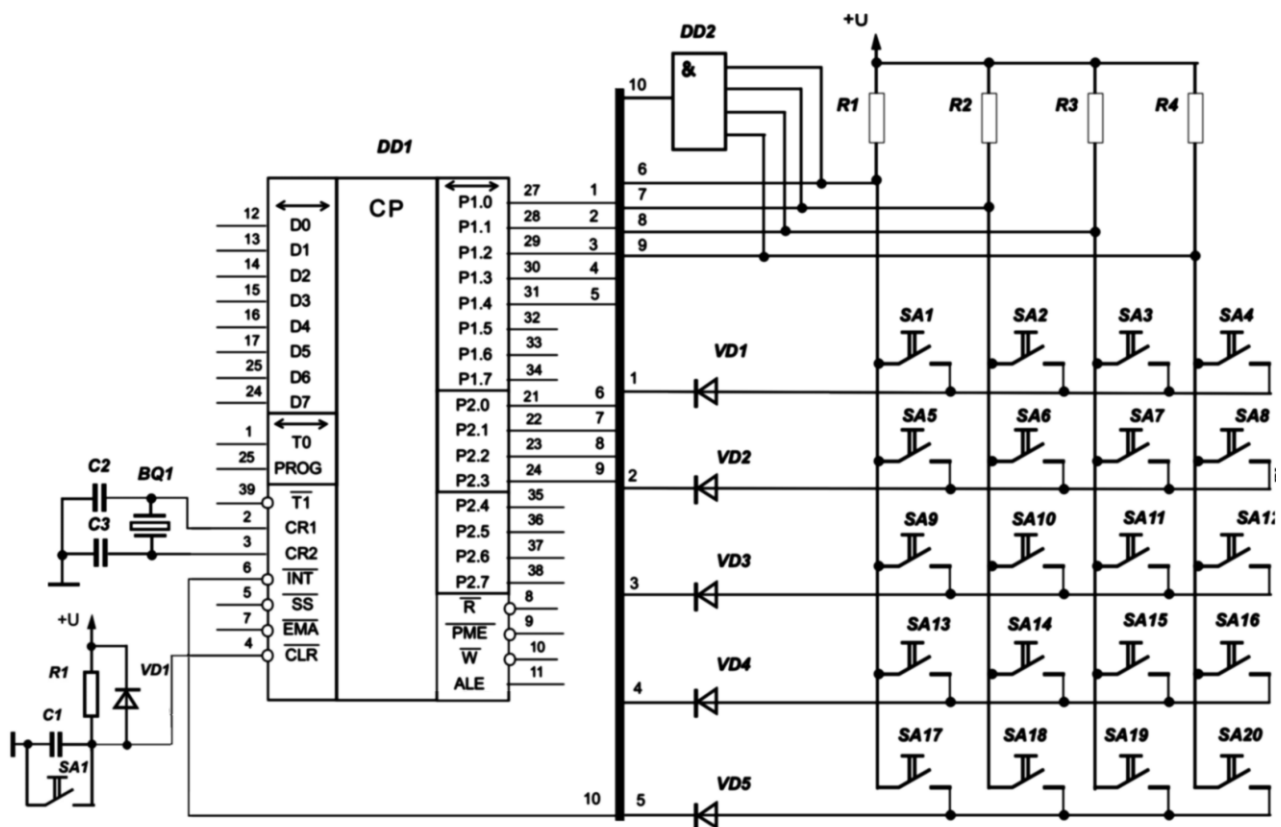


Рис. 56. Схема під'єднання клавіатури

Для деякого типу клавіатур може бути відсутня процедура усунення деренчання контактів. Процедуру сканування іноді буває зручно поєднати з процедурою ідентифікації. Розглянемо окремо кожен 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 залежно 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 ;ввід 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 введення 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, процедурі 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 перенесення, 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
              ;кнопки
CPL C        ;встановлення ознаки перенесення
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. Виведення і відображення інформації. Індикатори

Як пристрої виведення 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.

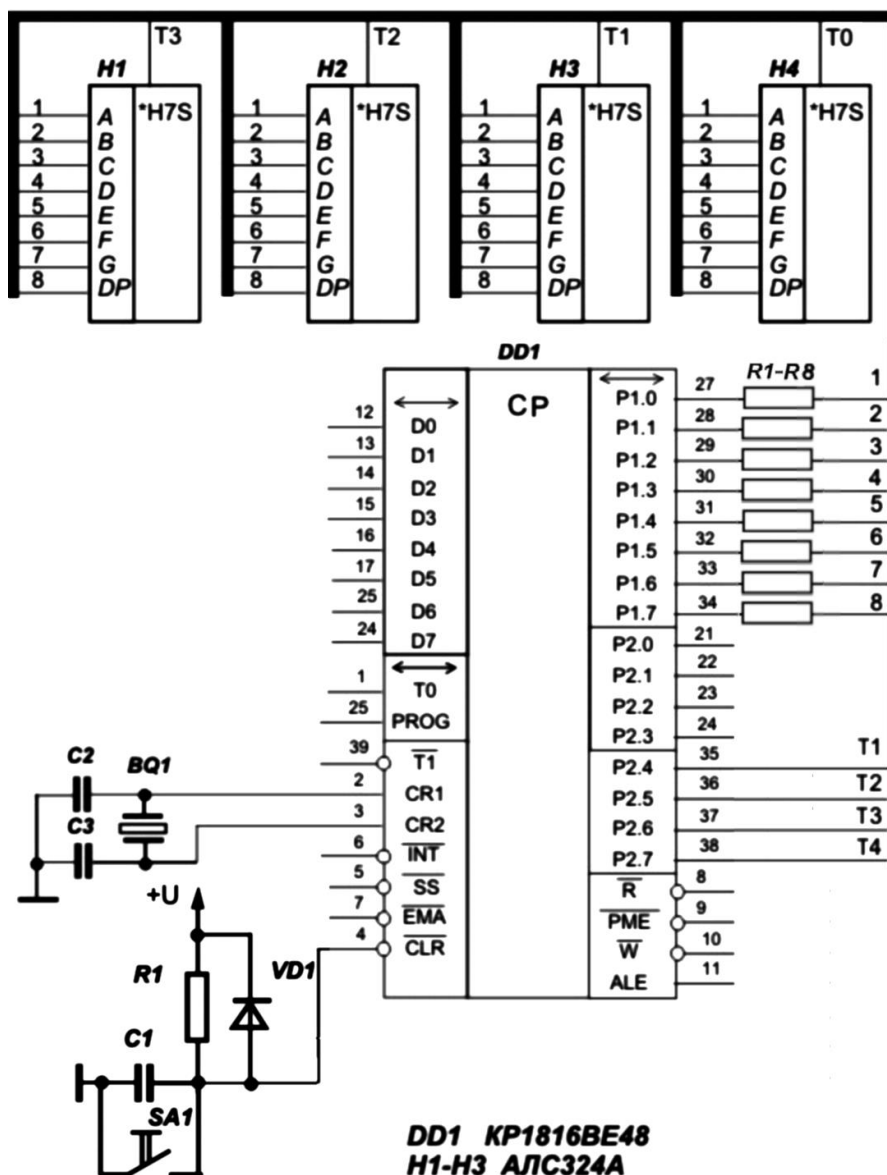


Рис. 57. Приклад під'єднання дисплея до КР1816ВЕ48

Багато МК-пристроїв вимагають наявності тільки найпростішої індикації на взірець 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 шістнадцяткові цифри, деякі літери кирилиці й 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).

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).

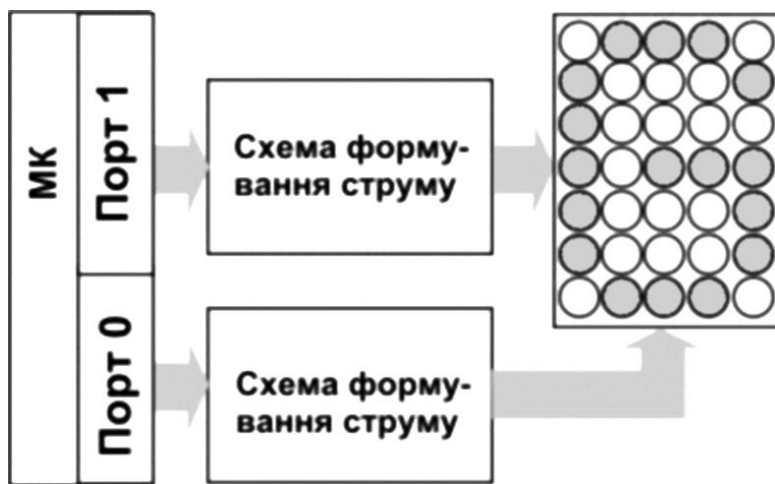


Рис. 58. Схема під'єднання матричного дисплея

Для ввімкнення одного світлодіода матриці необхідно забезпечити протікання через нього струму 10 – 15мА при напрузі 2,0 – 2,5В. 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 виведення інформації на дисплей вимагає значно менших апаратних витрат, але складнішого програмного забезпечення. Саме цей спосіб організації введення інформації набув переважного поширення в МК-системах.

3.5.1. Виведення 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 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  
MOVR3 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 CCI 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
```

повинен поставити у відповідність двійковому коду зображуваного символу визначений байт індикації (BI), який і

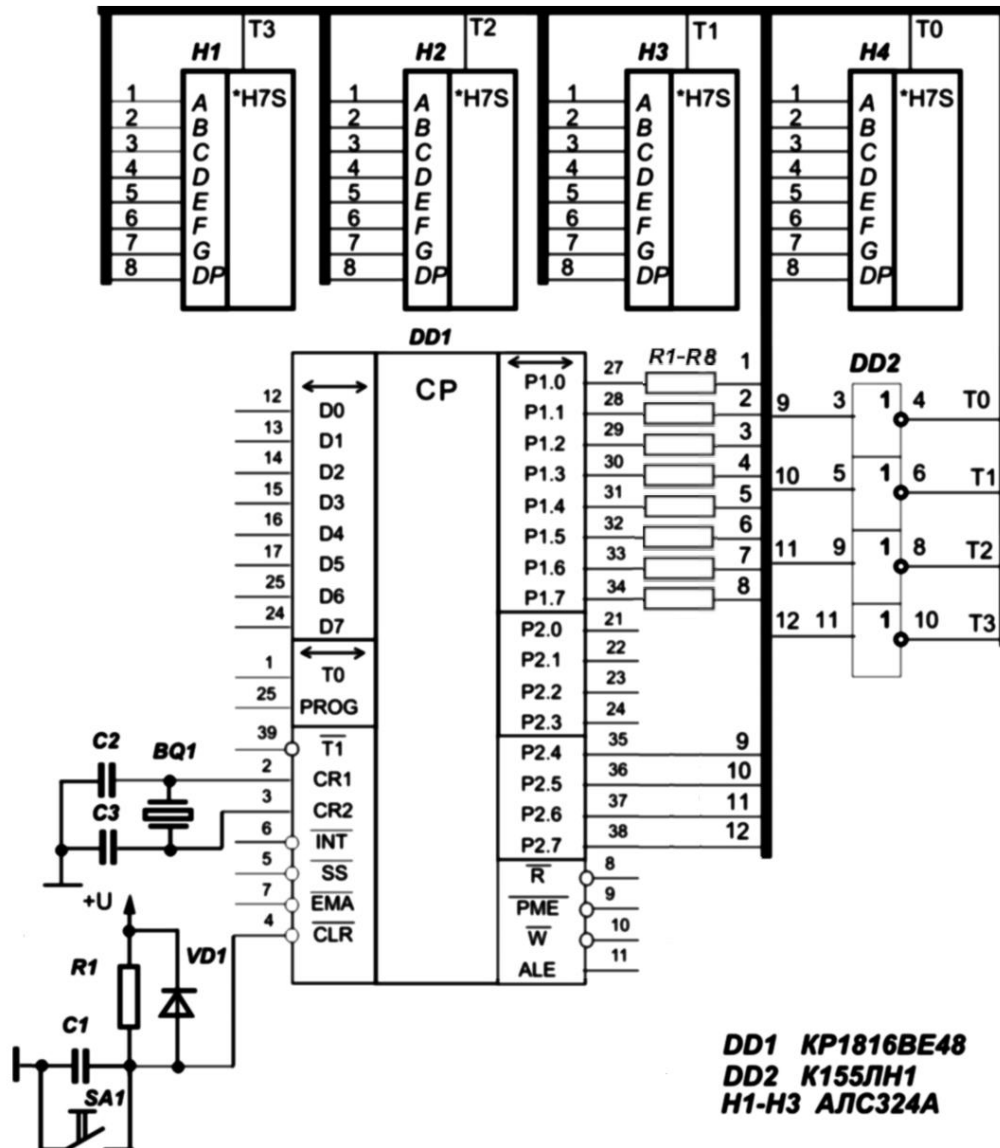


Рис. 59. Лінійний дисплей на семисегментних світлодіодних індикаторах

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 бітів
        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 й 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 знакомісці 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. Чекання 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.

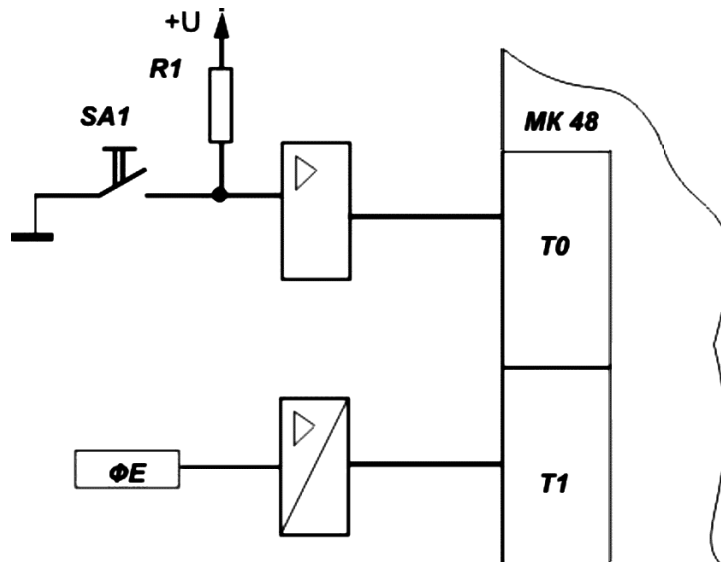


Рис. 60. Введення інформації з датчиків

3.6.2. Чекування статичного сигналу

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

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

Конкретна програмна реалізація процедури залежить не тільки від МК, але й 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 MCS-48).

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 MCS-48 програма буде мати вигляд

```

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

```

Інший варіант типової процедури чекування події є процедура чекування 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 й 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 повинен 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 передує 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

```

        JBZ WAITC ;якщо P1.3=0, то чекати
WAITO: IN A, P1 ;ввід байта
        JBZ WAITO ;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. Воно 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 протягом інтервалу часу.

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

Контакт вважається надійно замкненим (деренчання усунено), 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 наведено 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 ввімкнення 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: JT0 WAITC1 ;чекання ввімкнення конвеєра

```

```

WAITC2:  JT1 WAITC2  ;чекання початку імпульсу
WAITO2:  JNT1 WAITO2 ;чекання кінця імпульсу
          INC A;      ;інкремент лічильника деталей
          DA A        ;десятькова корекція
          JNT0 WAITC2 ;якщо конвеєр не вимкнений, то
                          ;продовжувати підрахунок
                          ;інакше - вихід із процедури

```

Після закінчення виконання процедури в акумуляторі буде знаходитися 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:  JT0 WAITC1  ;чекання ввімкнення конвеєра
          STRT CNT    ;запуск лічильника подій
          JNT0 WAITO1;чекання виключення конвеєра
WAITO1:  STOP TCNT   ;зупинка лічильника
          MOV A,T     ;зчитування вмісту лічильника
          . . . . .

```

В акумуляторі фіксується кількість деталей, представлена у двійковому коді (максимальна кількість деталей 255).

3.6.5.2. Підрахунок числа імпульсів за заданий проміжок часу

При вирішенні завдань перетворення числово-імпульсного коду в двійковий, а також у ряді інших завдань 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 апаратурний 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 за таблицею. 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 стану 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 клавішний регістр, процедура KBRD повинна бути доповнена процедурою ідентифікації натисненої клавіші й захисту від одночасного натискання двох і більше клавіш.

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

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 MCS-48. 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 MCS-48. Режими роботи. Передавання сигналів через порти.
9. Адресний простір пам'яті.
10. Під'єднання до MCS-48 зовнішньої пам'яті даних та програм.
11. Спряження MCS-48 з розширювачем вводу-виводу.
12. Під'єднання до MCS-48 розширеної кількості давачів. Під'єднання індикаторів та клавіатури.

ТЕМА №4. ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ НА БАЗІ ОДНОКРИСТАЛЬНИХ МІКРОЕОМ СІМЕЙСТВА MCS-51

4.1. Однокристалні мікроЕОМ сімейства MCS51. Основні характеристики

ОМЕОМ сімейства MCS-51 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

Система команд ОМЕОМ 87C51 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 ЗУПІ програми.

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

внутресхемного програмування тощо.

Базовий кристал сімейства i80C51 має такі характеристики:

- ◇ об'єм резидентної пам'яті програм – 4 кбайт;
- ◇ тип резидентної пам'яті програм – репрограмоване ПЗП РПЗП;
- ◇ об'єм резидентної пам'яті даних – 128 байт;
- ◇ мінімальна частота тактових сигналів 1,2 МГц;
- ◇ максимальна частота тактових сигналів 12 МГц;
- ◇ напруга живлення, +5 В ± 10%;
- ◇ струм споживання, 8 мА;
- ◇ обсяг зовнішньої адресованої пам'яті програм, 64 кбайт;
- ◇ обсяг зовнішньої адресованої пам'яті даних, 64 кбайт;
- ◇ система команд ОМЕОМ i80C51 містить 111 базових команд;
- ◇ багаторівнева система переривань підтримує переривання від п'яти джерел.
- ◇ Мікросхема має два зовнішніх входи запитів переривань.

Дуже корисна особливість i80C51 – обробка бітових даних, що дозволяє використовувати бінарну логіку, котра оперує бітами внутрішнього ОЗП та регістрів. Ця особливість широко використовується у промисловій автоматичі. Ще одна корисна особливість – чотири незалежні набори регістрів. Які дуже значно зменшують затримки при обслуговуванні переривань.

До складу ОМЕОМ i80C51 входять такі додаткові пристрої:

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

У позначенні ОМЕОМ МК-51 цифри мають таке призначення:

- ◇ перша цифра показує розрядність АЛП (8 біт);

- ◇ друга цифра задає тип внутрішньої пам'яті програм: 0 – ПЗП масочного типу, 3 – ПЗП з зниженою напругою живлення (+3В); 7 – репрограмоване з ультрафіолетовим стиранням, 9 – репрограмоване ПЗП з електричним стиранням,
- ◇ літера указує технологію виготовлення: С – пристрій, виконаний за КМОН технологією, відсутність літери – указує n-МОН технологію;
- ◇ останні дві цифри задають код сімейства: 51 – MCS-51 (MCS51), 31 – варіант без внутрішнього ПЗП програм.

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

- ◇ D – керамічний корпус DIP 40 виводів;
- ◇ P – пластиковий корпус DIP 40 виводів;
- ◇ N – корпус PLCC, 44 виводи.

Керамічний корпус DIP має вбудоване вікно, закрите кварцовим склом, для стирання ультрафіолетовим випромінюванням записаної в пам'ять програм. Пластиковий корпус DIP вікна не має, тому OMEOM P87C51 вважається однократно програмованим виробом, призначеним для дрібносерійного виробництва. Випускають також мікросхеми з підвищеною швидкодією – із граничним значенням тактової частоти 16, 20 і 24 МГц. Крім базового виробу, до складу сімейства MCS-51 входять такі модифікації:

- ◇ 87C52, 87C54 і 87C58, котрі мають внутрішнє ОЗП даних ємністю 256 байтів, пам'ять програм ємністю відповідно 8, 16 та 32 кбайти, а також додатковий лічильник-таймер;
- ◇ 87C51FA, 87C51FB, 87C51FC, аналогічні за характеристиками, 87C52, 87C54, 87C58, та мають у своєму складі кілька програмованих лічильників-таймерів. Таймери забезпечують функції типу ШІМ, прискореного послідовного виведення та інш.;
- ◇ 87C51GB, що має ОЗП даних ємністю 256 байтів та ПЗП ємністю 8 кбайтів, 8 аналогових входів та 16 рівноцінних портів вводу-виводу.

З OMEOM сімейства MCS-51, які випускають інші фірми, на особливу увагу заслуговують вироби фірми Atmel:

- ◇ AT89C51, яке містить ПЗП з електричним стиранням. Мікросхема має підвищену навантажувальну здатність портів, та забезпечує пряму роботу з світлодіодами:
- ◇ AT89C52, аналогічні 87C52, але з ПЗП з електричним стиранням;
- ◇ AT89C1051 і AT89C2051, що мають ПЗП з можливістю програмування ємністю, 1 чи 2 кбайти, вбудований компаратор аналогових сигналів, один чи два лічильники-таймери. Особливою відмінністю мікросхем є скорочене число виводів. Мікросхеми мають 15 програмованих ліній вводу-виводу з підвищеною навантажувальною здатністю.

Усі вироби фірми Atmel працюють при живленні від джерела напруги 2,7...6 вольт і при зміні частоти тактового генератора, що задається в діапазоні від 0 до 24 МГц.

Серед інших виробів сумісних з МК-51 варто відзначити сімейство C8051F12x фірми Cygnal. Сімейство C8051F12x – це i80C51 сумісні мікроконтролери з дуже високою продуктивністю, що досягає 108 операцій у секунду, великим об'ємом пам'яті та вбудованою аналого-цифровою периферією. На відміну від базових мікросхем, ядро C8051F12x виконує одну операцію за один такт та працює при тактовій частоті 100 МГц. Тобто еквівалентна тактова частота C8051F12x дорівнює 1.2 ГГц. Розглянемо "класичний" контролер сімейства MCS51 KM1816BE51.

Також варто відмітити споріднене сімейство MCS-52. На відміну від MCS-51, мікро- EOM сімейства MCS-52 мають:

- ◇ вбудоване ПЗП більшого об'єму;
- ◇ додаткові спеціальні функціональні регістри;
- ◇ третій лічильник-таймер, здатний працювати в режимах захоплення, лічильника, який допускає підрахунок як на збільшення,

так і на зменшення, і може працювати як генератор швидкості передачі послідовного порта;

◇ розширений програмований послідовний інтерфейс з детектуванням помилок передачі і автоматичним розпізнаванням адреси;

◇ розширений режим зниження споживаної потужності.

Мікро-ЕОМ MCS-52 використовують стандартний набір команд сімейства MCS-51, їх виводи взаємно відповідають. Відмінність полягає лише в тому, що, крім вводу/виводу інформації виводи порта P1 MCS-52 можуть виконувати альтернативні функції пов'язані із роботою додаткового лічильника-таймера

◇ перший з них використовується як зовнішній вхід для Т/С2,

◇ другий керує перезавантаженням /запам'ятовуванням інформації в регістри Т / С2.

4.2. Архітектура ОМЕОМ 80С51

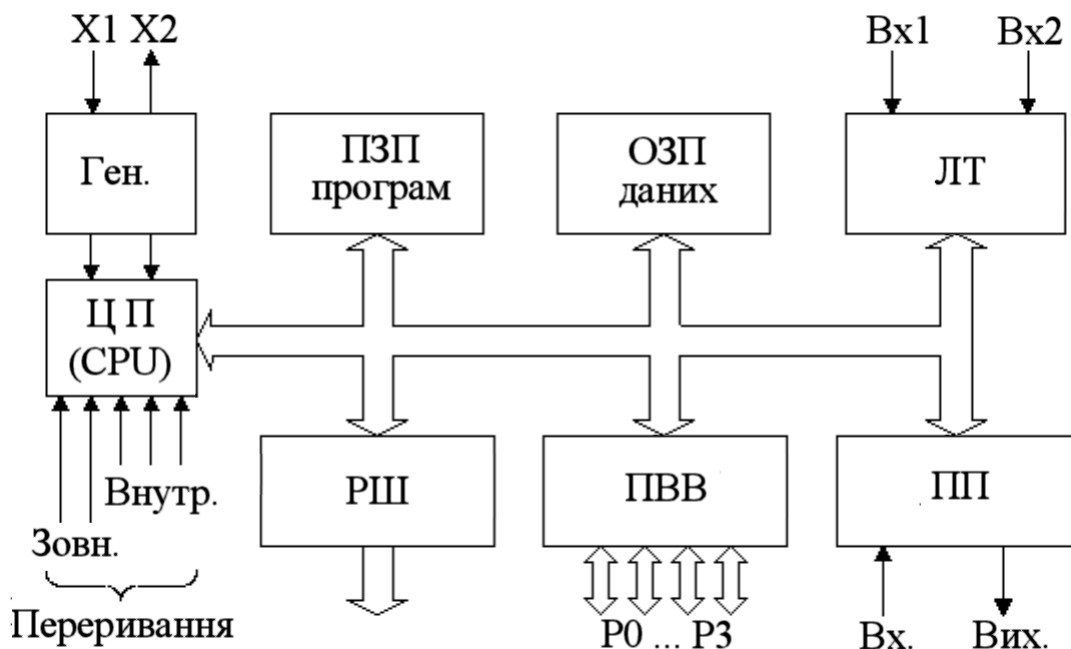


Рис. 61. Архітектура мікроЕОМ

Пам'ять програм і пам'ять даних в ОМЕОМ сімейства MCS-51 не тільки фізично й 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 акумулятора (A), регістр B, 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 выводами 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.

Введення 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 виведення 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 якогось завдання 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 (введення кодів команд, введення\виведення 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.

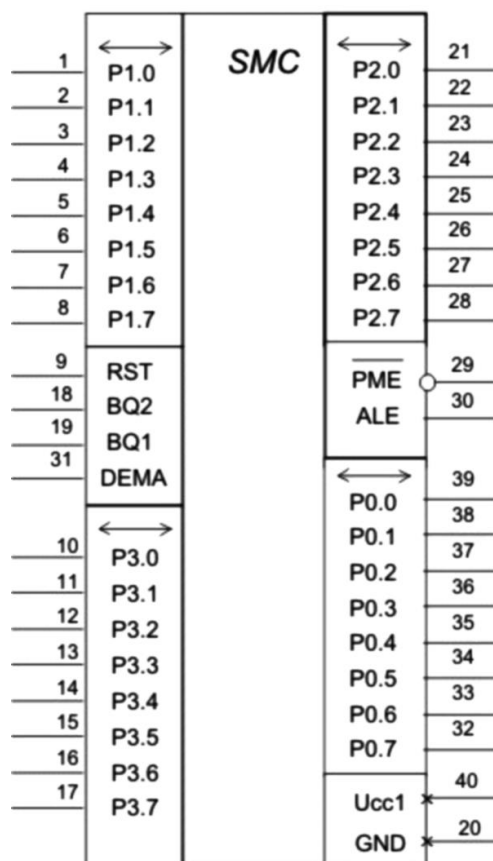


Рис. 62. Позначення КР1816ВЕ751

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 зі входами й 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 OMEOM був заздалегідь виставлений лог. 1. У протилежному випадку лінія буде "посажена" 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 запису 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

1	2	3
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
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 та обміну даними
Vcc1	40	Напруга живлення 5В

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

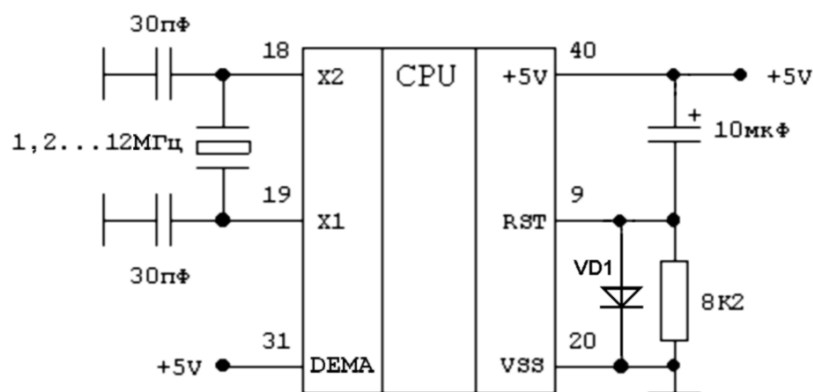


Рис. 63. Мінімальна схема ввімкнення ОМЕОМ 87С51

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 однокристального 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

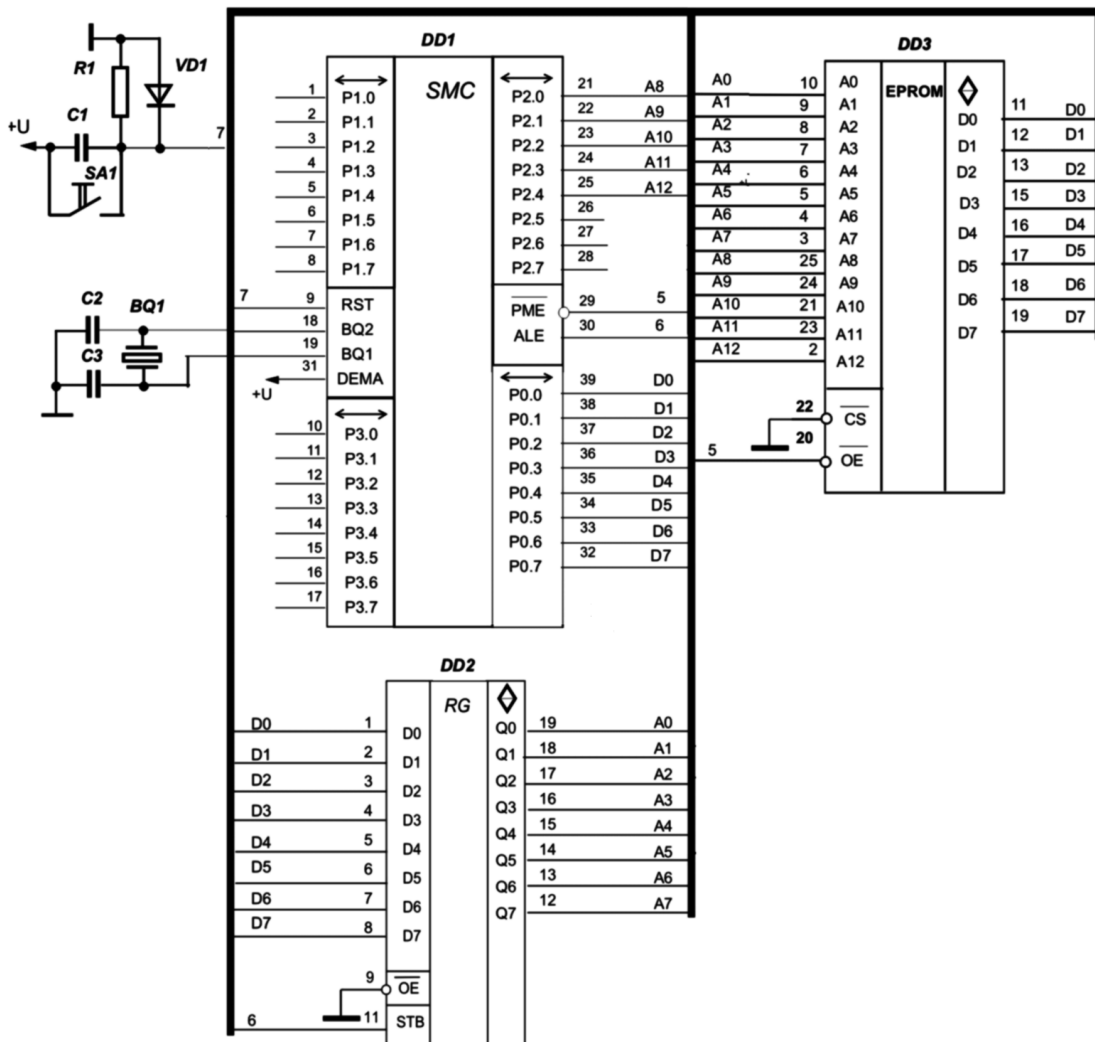


Рис. 64. Під'єднання ПЗП

На рис. 64 зображено структуру МК-системи з зовнішньою пам'яттю програм. Шина P0 за своїми властивостями подібна до двонаправленої шини даних мікропроцесора KP580 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);
- ◇ за зрізом сигналу ALE на зовнішньому регістрі фіксується адреса;
- ◇ сигналом PМЕ дозволяється робота зовнішньої пам'яті;
- ◇ за спадом сигналу PМЕ шина P0 переходить у режим введення.

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) MCS-51.

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 із використанням 8-бітної адресації не змінює стан порту P2, отже, вимагає попереднього встановлення потрібного значення старших розрядів адреси за допомогою порту P2.

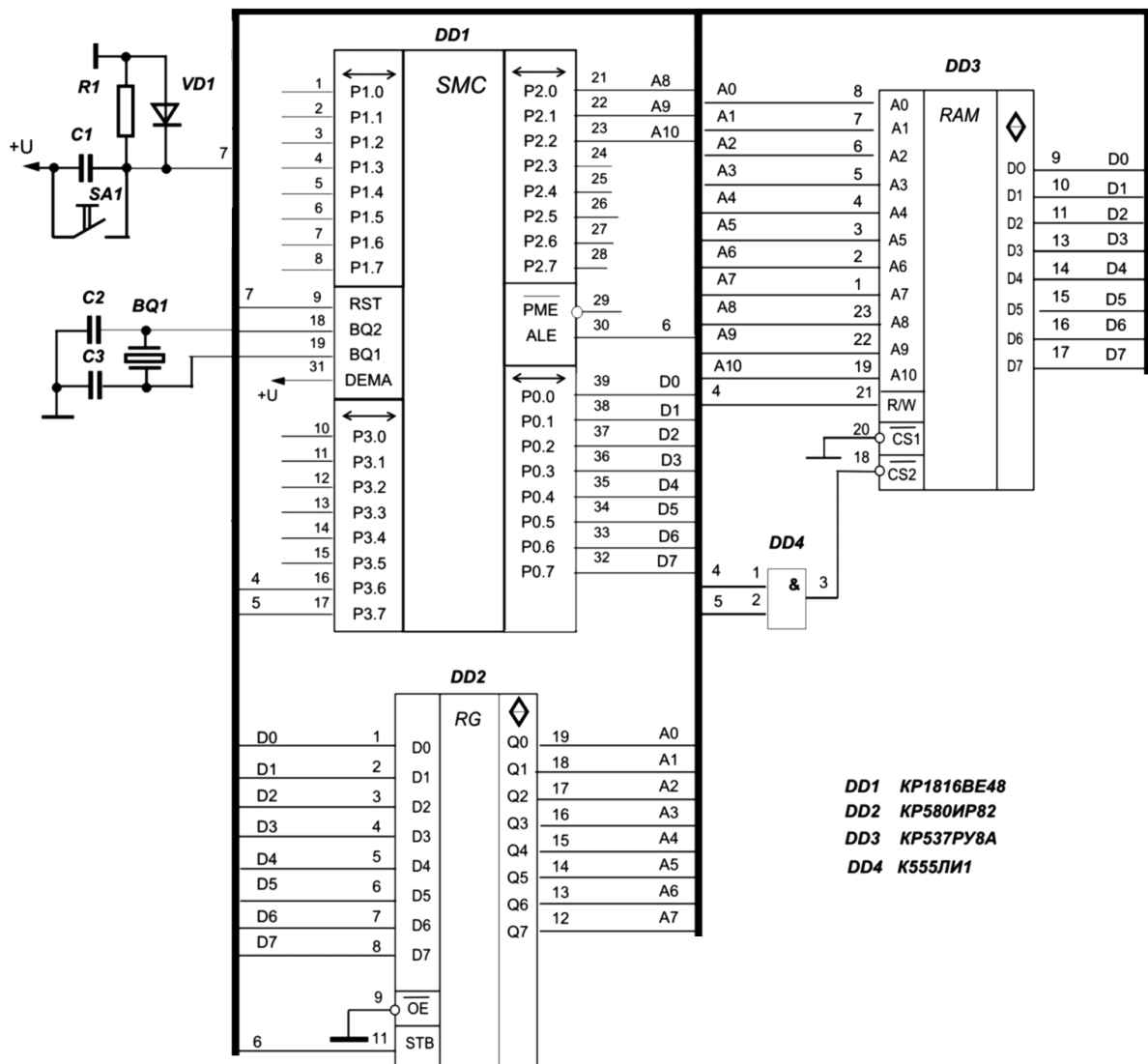


Рис. 65. Розширення пам'яті даних KP1816BE51

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 MCS-48, можна виконати з використанням стандартного розширювача вводу-виводу (РВВ) КР580ВР43, або інтерфейсних ВІС (КР580ВВ55, КР580ВВ51).

Розширювач під'єднується до MCS-51 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.

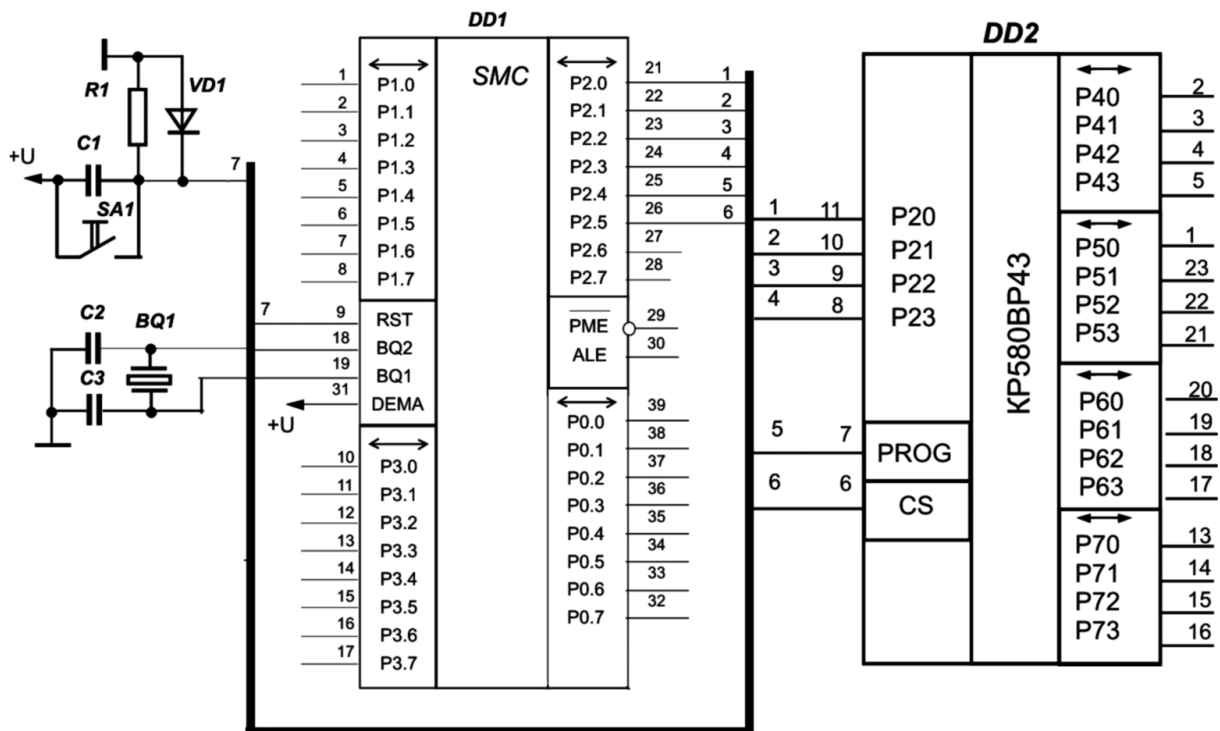


Рис. 66. Під'єднання розширювача вводу-виводу

Розширення ПВВ за допомогою ППІ серії КР580 здійснюється аналогічно схемам для MCS-48, зображених на рис. 55. Прикладом може бути схема, що на рис. 67.

Мікросхема КР580ВВ55 для мікроЕОМ відображається на комірки 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 A2-A7, 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 MCS-51 виконується за тими ж правилами, що і для схем на основі MCS-48.

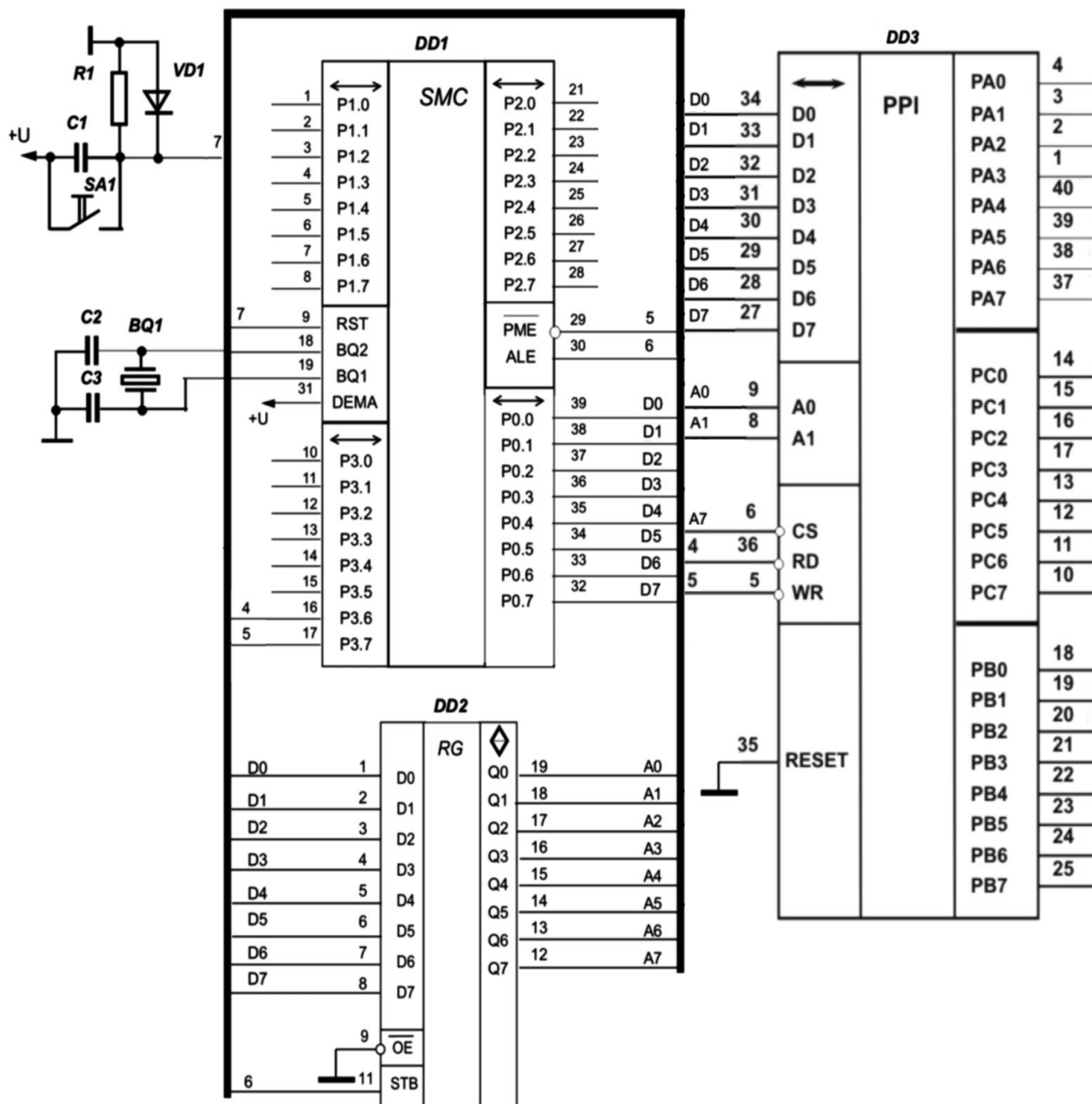


Рис. 67. Приклад під'єднання мікросхем серії КР580 до мікроЕОМ

Організація більшості програмних процедур для MCS-51 також є такими, як і для MCS-48. З урахуванням того, що в MCS-51 кожен вивід кожного порту

може бути перевірений однією командою без необхідності завантаження стана порту в акумулятор.

4.6. Лічильник-таймер у MCS-51

Два програмованих 16-бітних лічильники/таймери (T/C0 і T/C1) можуть бути використані в якості таймерів або лічильників зовнішніх подій. При роботі в якості таймера вміст T/C інкрементується у кожному машинному циклі, тобто через кожні 12 періодів резонатора. При роботі в якості лічильника вміст T/C 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 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.

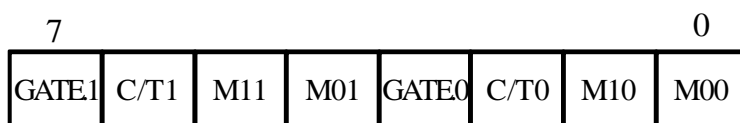


Рис. 68. Формат регістра режиму TMOD

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 "Тх"
M1.x, M0.x	Режим роботи, що вибирається згідно з таблицею 29

Старший та молодший байти лічильника носять назву ТНх та ТЛх. Керування лічильниками здійснюється загальним регістром режиму TMOD та регістром конфігурації TCON, стан лічильників відображається у регістрі TCON. Опис регістрів наведено в таблицях 28 та 30, структура регістрів зображена на рис. 68 та 69.

Таблиця 29. Режим роботи таймера залежно від бітів настроювання

M1	M0	Режим роботи
0	0	Таймер сумісний з MCS-48. ТЛ працює як 5-бітний попередній подільник, ТН – у режимі, сумісному з таймером K1816BE48
0	1	16-бітний лічильник/таймер. ТЛ і ТН ввімкнені послідовно
1	0	8-бітний лічильник/таймер, що перевантажується. ТН зберігає значення, що повинно бути перезавантажене в ТЛх у момент переповнення
1	1	Таймер/лічильник 1 зупиняється. У таймері/лічильнику 0 ТЛ0 працює як 8-бітний таймер/лічильник, і його режим визначається керуючими бітами таймера 0. ТН0 працює тільки як 8-бітний таймер, і його режим визначається керуючими бітами таймера 1

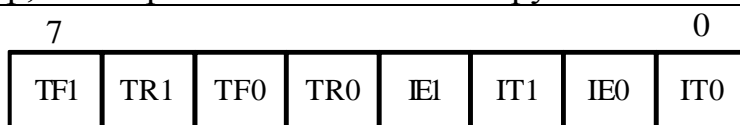


Рис. 69. Формат керування/статусу таймера

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. 70а). 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 або керуючий біт GATE дорівнює 0, або на зовнішній вивід запиту переривання INT1 надходить рівень 1.

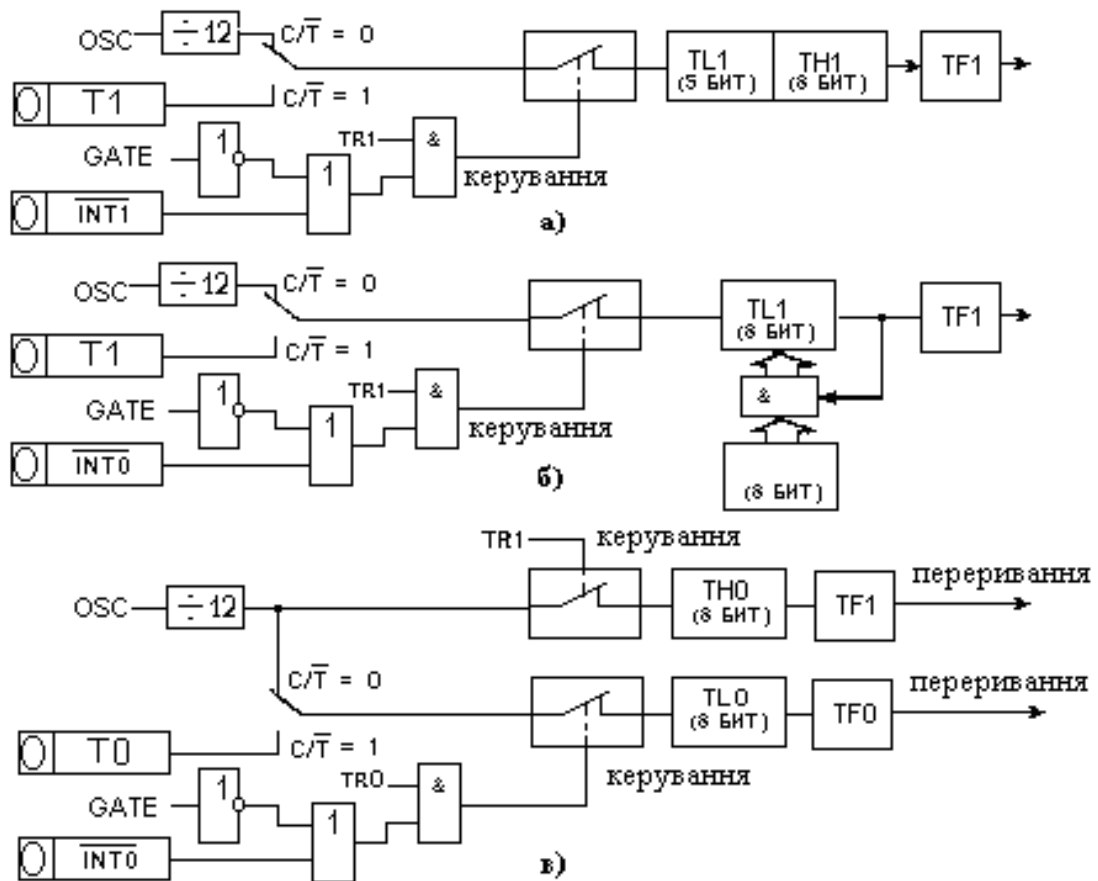


Рис. 70. Таймер/лічильник подій:

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 Т/С 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 зі стану "усі одиниці" у стан "усі нулі") 8-бітного лічильника TLx призводить не тільки до встановлення ознаки TF, але й автоматично перезавантажує у TLx уміст старшого байта (ТНх) таймерного регістра, що попередньо було задано програмним шляхом. Перезавантаження залишає вміст ТНх 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 уміст, іншими словами, ефект такий же, як і при скиданні керуючого біта TRI 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 INTO 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 випадку, якщо лічильник/таймер 0 використовується в режимі 3, лічильник/таймер 1 може бути або ввімкнений, або вимкнений, або переведений у 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 MCS-51 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. Запис 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 бітів інформації: старт-біт (0), 8 бітів даних і стоп-біт (1). Швидкість приймання/передавання змінна і задається таймером.

У режимі 2 через TXD передаються або з RXD приймаються 11 бітів інформації: старт-біт, 8 бітів 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 вміщене значення ознаки паритету зі 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).

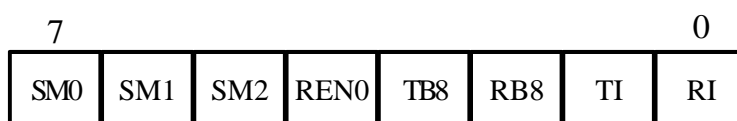


Рис. 71. Формат регістра керування та статусу УАПП (SCON)

Формат регістра, що керує роботою УАПП, зображено на рис. 71. Призначення бітів регістра керування/статусу УАПП наведено 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 задавання 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 закінчення передавання байта. Скидається програмно після обслуговування переривання
RI	Ознака переривання приймача. Установлюється апаратно 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 УАПП у мультимікроконтролерних системах

У системах децентралізованого керування, що використовуються для керування й регулювання в розподілених об'єктах (наприклад, прокатних станах, рухомому складі залізниці й 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 УАПП дозволяє відносно простими засобами реалі-

зувати 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. УАПП 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 моно каналу, 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.



Рис. 72. Формат регістра спеціальних функцій PCON

У режимі 0 швидкість обміну залежить лише від резонансної частоти кварцового резонатора ($f_{рез}$) і дорівнює $f_0 = f_{рез}/12$. – 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).

Таблиця 32. Біти регістра спеціальних функцій

Символ	Ім'я і призначення
SMOD	Подвоєна швидкість обміну. Якщо біт встановлений в 1, то швидкість обміну вдвічі більша, 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

У 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:

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

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

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

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
------------------	--------------	-------------------------	------	------------------------------------

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 у 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, встановлює 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").

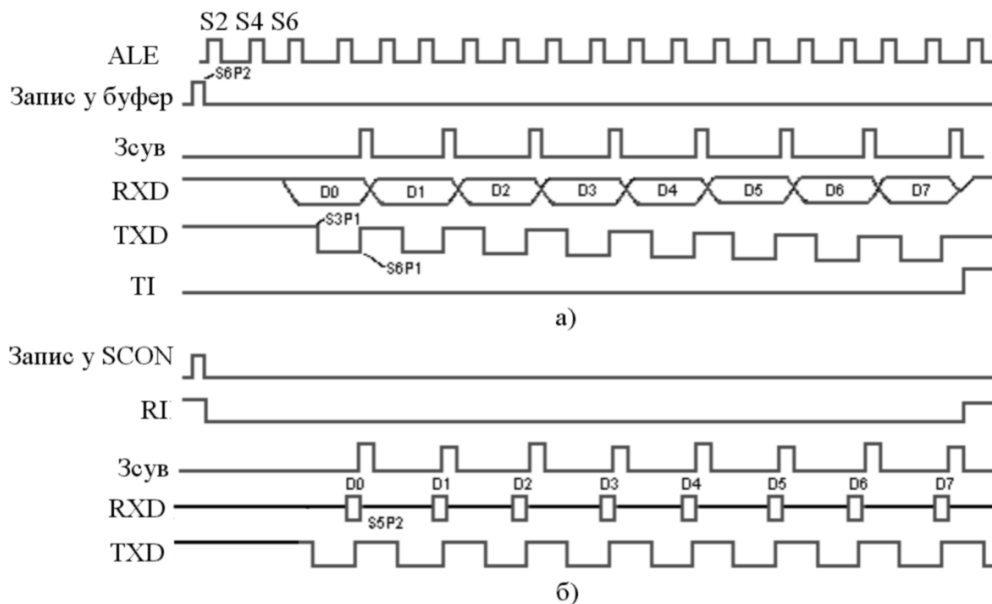


Рис. 73. Часова діаграма обміну в режимі 0:

а) передавання даних, б) приймання даних

Приймання починається за умови $REN = 1$ і $RI = 0$. У 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 у 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 сигнал, що дозволяє прийом, і встановлює прапорець RI.

4.7.5.2. Режим 1

Часова діаграма роботи УАПП у режимі 1 зображена на рис. 74. Через вивід TXD УАПП передає, а вивід RXD приймає 10 бітів: старт-біт, 8 бітів даних, можливо, дев'ятий біт даних і стоп-біт. При прийманні стоп-біт надходить у біт RB8 регістра SCON.

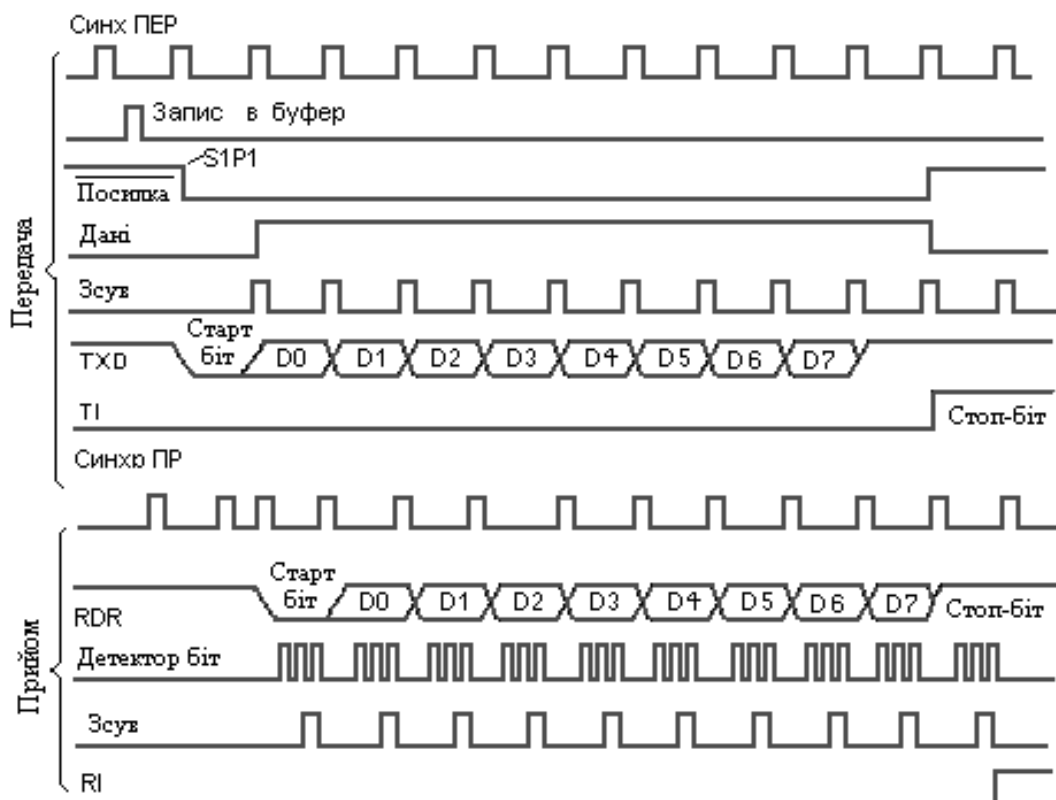


Рис. 74. Часова діаграма роботи УАПІ у режимі 1

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. За цим сигналом на вивід TXD спочатку надходить старт-біт, а потім за сигналом, що дозволяє, – 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

Через вивід TXD УАПП 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 УАПП 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.

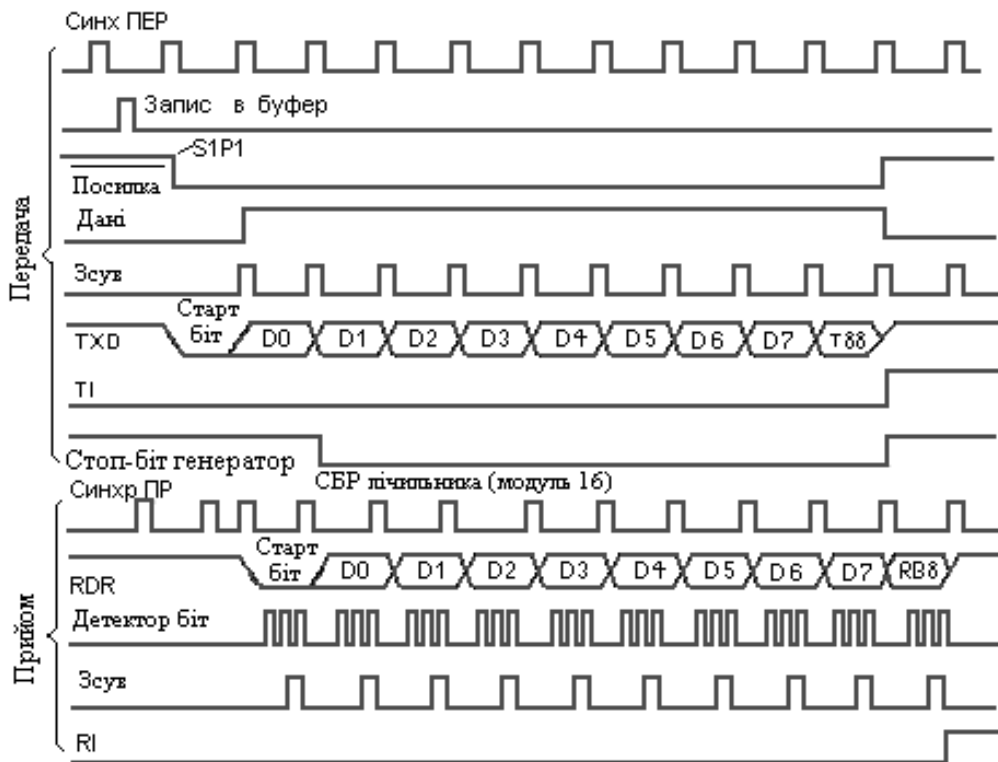


Рис. 75. Часова діаграма роботи УАПІ у режимах 2 і 3

4.8. Система переривань

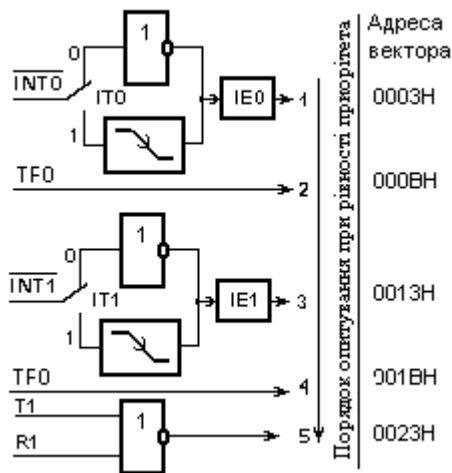


Рис. 76. Схема переривань MCS51

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 устанавлюються 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 викликано рівнем вхідного сигналу, то скиданням прапорця ІЕ керує відповідна підпрограма обслуговування переривання шляхом впливу на джерело переривання.

Ознаки запитів переривання від таймерів TF0 і TF1 скидаються автоматично при передаванні керування підпрограмі 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 УАПП. 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 зображено на рис. 78, опис

розрядів наведено в таблиці 35.

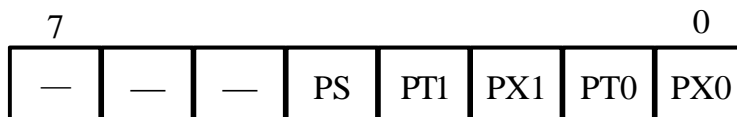


Рис. 78. Формат регістра пріоритетів переривань IP

Таблиця 35. Регістр пріоритетів переривань IP

Символ	Ім'я і призначення
PS	Біт пріоритету УАПП. Одиниця вказує на високий пріоритет переривання УАПП, нуль – на низький
PT _x	Біт пріоритету таймера x. Одиниця вказує на високий пріоритет переривання таймера, нуль – на низький
PX	Біт пріоритету зовнішнього переривання (INT _x). 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 зі таких умов:

- 1) у даний момент обслуговується запит переривання рівного або вищого 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 обслуговувалося 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 у 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 й 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 IE.7 встановлено.

```

;організація переходу до мітки
;NEXT при переповненні
;лічильника/таймера 0
ORG 0BH ;адреса вектора преривання 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 THO, #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 MCS-51 із 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
WAIT0:
JNB P3.2, WAIT0 ;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 MCS-51 зовнішньої пам'яті даних та програм.
10. Проектування систем керування на базі MCS-51. Спряження MCS-51 з розширювачем вводу-виводу.
11. Проектування систем керування на базі MCS-51. Під'єднання до MCS-51 розширеної кількості датчиків.
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 всередині 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 внутрішнього 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 й опрацьовують 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 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 1024x14, в якості якої використовується PROM (в PIC16C84), FLASH (в PIC16F84) або масочний постійний запам'ятовуючий пристрій (в PIC16CR84);
- ◇ пам'яті даних об'ємом 36x8 (в PIC16F83, PIC16CR83, PIC16C84), або 68x8 (в PIC16CR84 та в PIC16F84), що може використовуватись в якості регістрів загального призначення;
- ◇ EEPROM даних об'ємом 64 байти, 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 B;
- ◇ після завершення запису 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

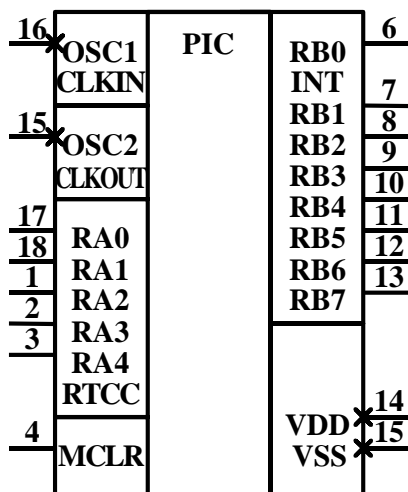


Рис. 79. Умове позначення мікросхеми

Умове позначення мікросхеми наведено на рис. 79, а призначення виводів – у таблиці 37.

Таблиця 37. Призначення виводів мікросхеми

Позначення	Номер виводу	Призначення
1	2	3
RA0-RA3	17, 18, 1, 2	Двонаправлені лінії вводу-виводу. Вхідні та вихідні рівні відповідають TTL

1	2	3
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 стана 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 ввімкнення 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

1	2
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 порту А	80 мА
Сумарний струм, що видається через лінії порту А	50 мА
Сумарний струм, що приймається через лінії порту В	150 мА
Сумарний струм, що видається через лінії порту 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 й 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.

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.

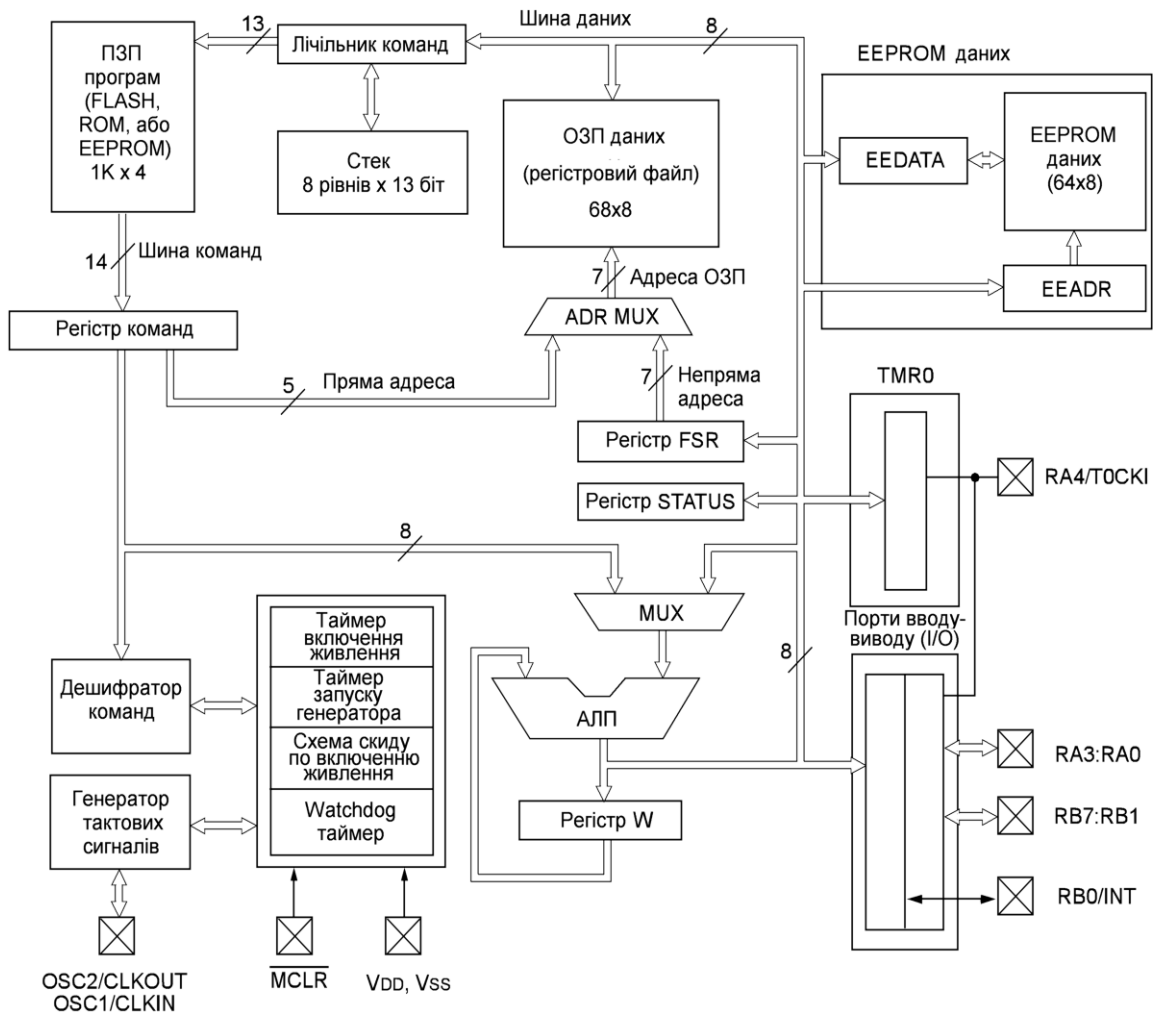


Рис. 80. Структурна схема мікросхеми

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 й керування іншими блоками мікросхеми. Він складається із двох таких 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 увійде 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 вимкнення таймера ТВЖ.

4. Watchdog таймер (WT), називають також сторожовим таймером – це спеціальний таймер, що приводить мікроЕОМ у 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.

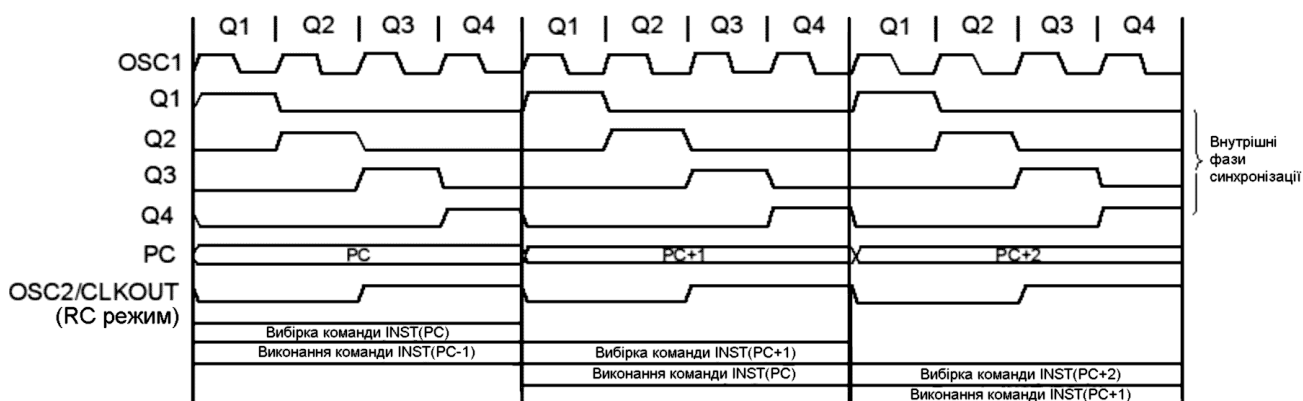


Рис. 81. Схема тактування і виконання команди

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 регістра 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 й 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 (00h-FFh), 1 = банк 2, 3 (100h-1FFh). Біт IRP не використовується в МК підгрупи PIC16F8X
RP1- RP0	Біти вибору сторінки банку даних (використовуються при прямій адресації): 00 = банк 0 (00h-7Fh); 01 = банк 1 (80h-FFh); 10 = банк 2 (100h-17Fh); 11 = банк 3 (180h-1FFh). У МК підгрупи PIC16F8X використовується тільки біт RP0
\overline{TO}	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

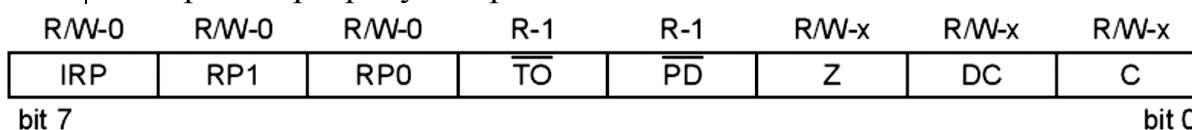


Рис. 82. Формат регістра STATUS

На рис. 82 прийняті такі позначення: R – біт, що читається; W – запису 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, запис 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 бітів \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 бітів 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. Зазвичай за адресою 0004H розташовується підпрограма ідентифікації й опрацювання переривань, а за адресою 0000H – команда переходу на програму початкового встановлення, розташовану за підпрограмою обробки переривань. Уся пам'ять програм є внутрішньою. Запустити програму з зовнішньої пам'яті неможливо.

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 x 14 пам'яті (адреси 0000h-01FFh), а в МК PIC16C84,

PIC16F84 і PIC16CR84 - 1К×14 пам'яті (адреси 0000h-03FFh). Звертання до адрес вище 1FFh (3FFh) є адресацією перших 512 або 1К комірок пам'яті програм.

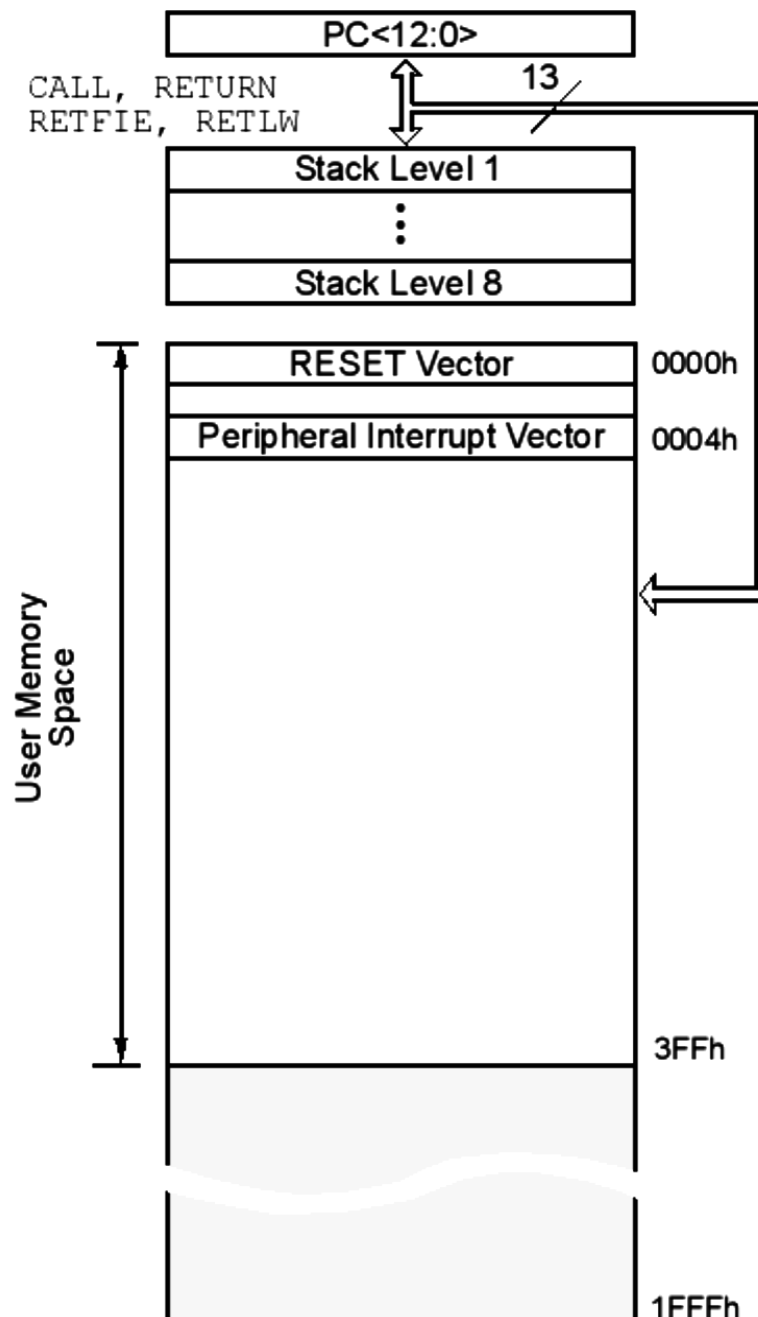


Рис. 83. Організація пам'яті програм і стека

Молодший байт лічильника (PCL) доступний для читання й запису і відображається на регістр з адресою 02h. Старший байт лічильника команд не може бути прямо записаний чи зчитаний і береться з регістра PCLATH (PC latch high), адреса якого 0Ah. 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.

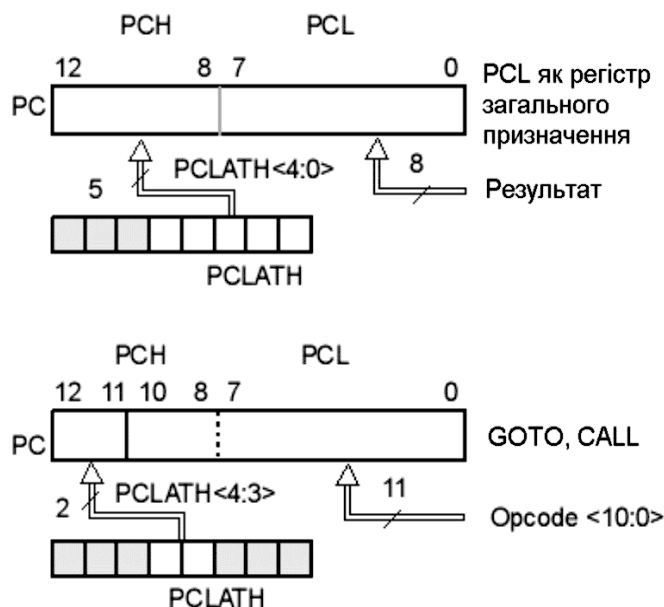


Рис. 84. Завантаження старших бітів лічильника команд

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 повинен 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 вміщується 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 перепише 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 непрямої адресації звертатися до регістрів пам'яті чи даних. Усі регістри спеціальних функцій, включаючи лічильник команд, відображаються на пам'ять даних.

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

Обидві області, у свою чергу, поділені 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.

Адреса			Адреса
00h	Непряма адреса ⁽¹⁾	Непряма адреса ⁽¹⁾	80h
01h	TMRO	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	—	—	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 ⁽¹⁾	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>ОЗП даних банк 0</p> <p>68 (36) регістрів загального призначення</p> </div> <div style="text-align: center;"> <p>Відображається банк 0 адреси 0Ch-4Fh (0Ch-2Fh)</p> </div> </div>		8Ch
4Fh(2Fh) 50h(30h)			CFh (AFh) D0h (B0h)
7Fh	Банк 0	Банк 1	FFh

■ - Фізично відсутні комірки, читаються як 0.

(1) - Регістр даних для непрямої адресації.

Рис. 85. Організація пам'яті даних

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

Регістри з адресами 0СН-4FН (0СН-2FН) 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 8СН-СFН 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 звертатися напряму, використовуючи абсолютну адресу кожного регістра, чи за допомогою непрямої 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 з коду операції, а два біти покажчика сторінок (RP1, RP0) – з регістра статусу, як зображено на рис. 86.

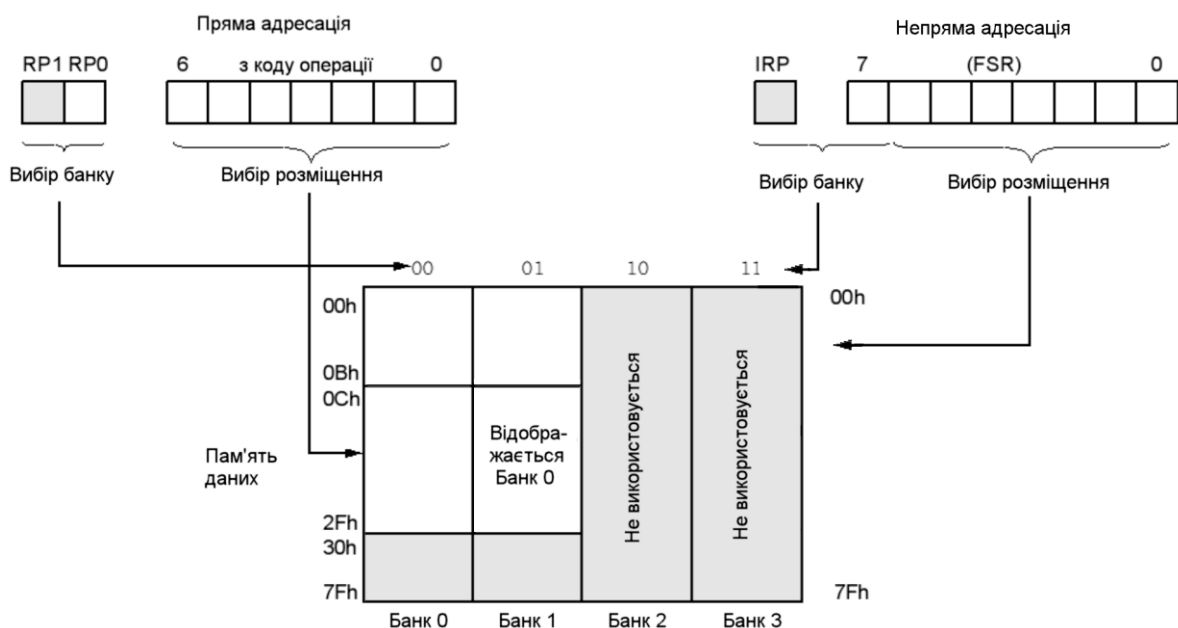


Рис. 86. Методи адресації даних

Ознакою непрямой адресації є звертання до регістра INDF. Будь-яка команда, що використовує INDF (адресу 00H) у якості операнда звертається до покажчика, що зберігається в FSR (адреса 04H). Непряме читання самого регістра INDF дасть результат 00H. Непрямий запис 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.

5.8. EEPROM даних PIC16X8X

Мікроконтролери підгрупи PIC16X8X мають енергонезалежну EEPROM пам'ять даних об'ємом 64x8 бітів, що допускає запис 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 непрямой адресації через регістри спеціальних функцій: EEDATA (з адресою 08H), через який здійснюється запис і читання даних, і EEADR (адреса 09H), котрий визначає адресу комірки, до якої здійснюється звертання. Для керування процесом читання та запису використовуються регістри: EECON1 (адреса 88H) і EECON2 (адреса 89H).

При записі байта в комірку EEPROM попереднє значення стирається і записуються нові дані. Всі операції зі стирання і запис 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 байтів даних EEPROM. У МК підгрупи PIC16X8X використовуються тільки перші 64 байти, 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 біти адресного простору.

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

Таблиця 40. Призначення бітів регістра EECON1

Біт	Значення
EEIF	Біт запиту переривання після закінчення 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 після завершення 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 після завершення читання і не може бути програмно скинутий в 0

Регістр EECON2 призначений для запобігання запису в EEPROM при збогах програми. Для дозволу запису в EEPROM перед записом 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 запису 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 записі в пам'ять EEPROM необхідно спочатку записати адресу в регістр EEADR і дані в регістр EEDATA. Потім необхідно виконати спеціальну послідовність команд, що виконує запис даних:

```

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 запису.

```

Під час виконання цієї ділянки програми всі переривання повинні бути заборонені для точного виконання тимчасової діаграми. Час запису – приблизно 10 мс. Фактичний час запису може змінюватися залежно від напруги, температури та індивідуальних властивостей мікросхеми. В кінці запису біт WR автоматично обнулюється, а ознака завершення запису 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 в 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 у цьому розділі не наведено. Виняток становить регістр конфігурації.

Таблиця 41. Регістри спеціальних функцій

Адреса	Назва	Призначення
1	2	3
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 непрямої адресації

1	2	3
05H	PORTA	Регістр, що вказує стан порту А. Значення мають лише молодші 5 бітів
06H	PORTB	Регістр, що вказує стан порту В
07H	-	Не використовується (зарезервований для порту С)
08H	EEDATA	Регістр даних для обміну з EEPROM
09H	EEADR	Регістр адреси для обміну з EEPROM
0AH	PCLATH	Старші 5 розрядів для запису 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 порту А
86H	TRISB	Регістр вказує напрямок передачі порту В
87H	-	Не використовується (зарезервований для порту 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)

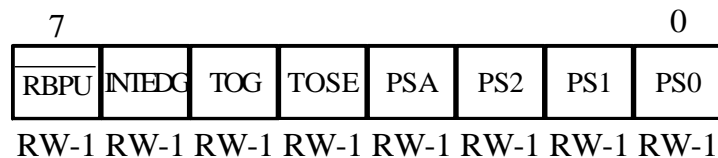


Рис. 88. Формат регістра конфігурації OPTION

Регістр конфігурації (OPTION) є доступним для читання і запису. Він містить керуючі біти для конфігурації попереднього подільника, зовнішніх переривань, таймера, а також підтягуючих ("pull-up") резисторів на виводах PORTB. Призначення біт-регістра наведено в таблиці 42, а формат регістра – на рис. 88.

Таблиця 42. Формат регістра конфігурації (OPTION)

Назва	Призначення
$\overline{\text{RBP}}\text{U}$	Біт установки підтягуючих «pull-up» резисторів на виводах PORTB: 0 – резистори під'єднані; 1 – резистори від'єднані

1	2		
INTEDG	Біт вибору переходу сигналу переривання: 0 – переривання по спаду сигналу на лінії RBO/INT; 1 – переривання по фронту 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 фронту 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 виведення.

Порт А (PORTA) – це 5-бітовий фіксатор, що відповідає виводам контролера RA<4:0>. Лінія RA4 має вхід тригера Шмітта і вихід з відкритим стоком. Усі інші лінії порту мають TTL вхідні рівні і КМОП вихідні буфери. Адреса регістра порту А – 05H.

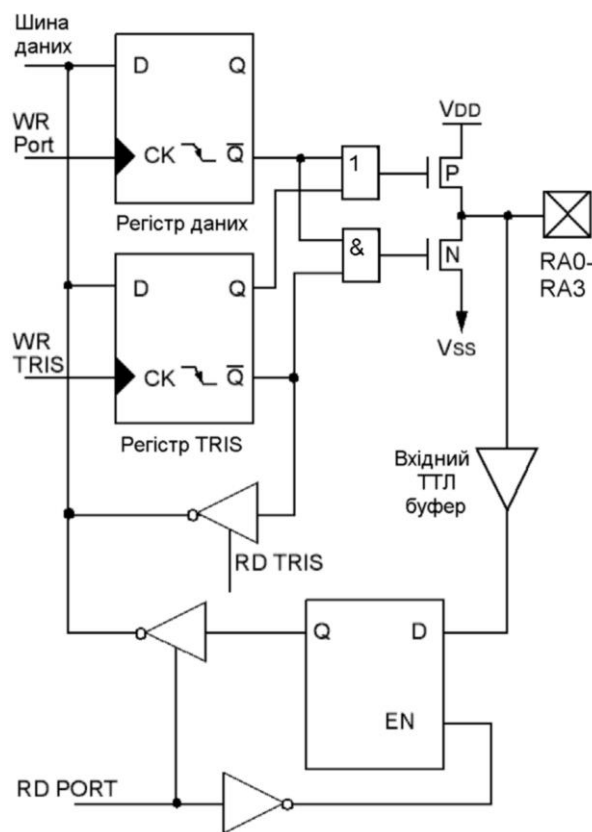


Рис. 89. Схема ліній RA<3:0> порту A

Кожній лінії порту поставлений у відповідність біт напрямку передавання даних у керуючому регістрі TRISA, розташованому за адресою 85H. 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 A. Зауважимо, що виводи порту мають захисні діоди до VDD і VSS.

Операція читання порту A 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 вихідної лінії, а не внутрішньої засувки. Отже, якщо в якийсь момент часу вихід буде тимчасово "посаджений" у нуль, цей нуль і буде зчитаним навіть коли лінія повинна виводити логічну одиницю.

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

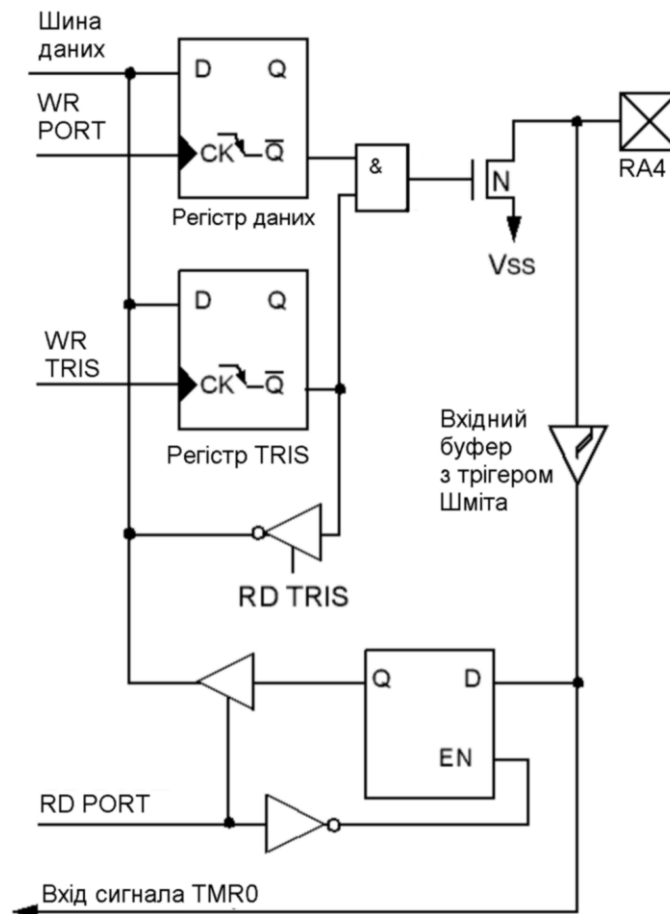


Рис. 90. Схема лінії RA4 порту А

Порт В (PORTB) – це двонаправлений 8-бітовий порт, що відповідає виводам RB<7:0> контролера і розташований за адресою 06Н. Кожній лінії порту поставлений у відповідність біт напрямку передавання даних, що зберігається в керуючому регістрі TRISB, розташованому за адресою 86Н. 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 під'єднане невелике активне навантаження (таке, що дає струм близько 10 мкА). Воно автоматично від'єднується, якщо цей вивід запрограмований як вихід. Керуючий біт $\overline{\text{RBPU}}$ регістра OPTION<7> може від'єднати (при $\overline{\text{RBPU}} = 1$) всі навантаження. Скидання при вмиканні живлення також вимикає всі навантаження.

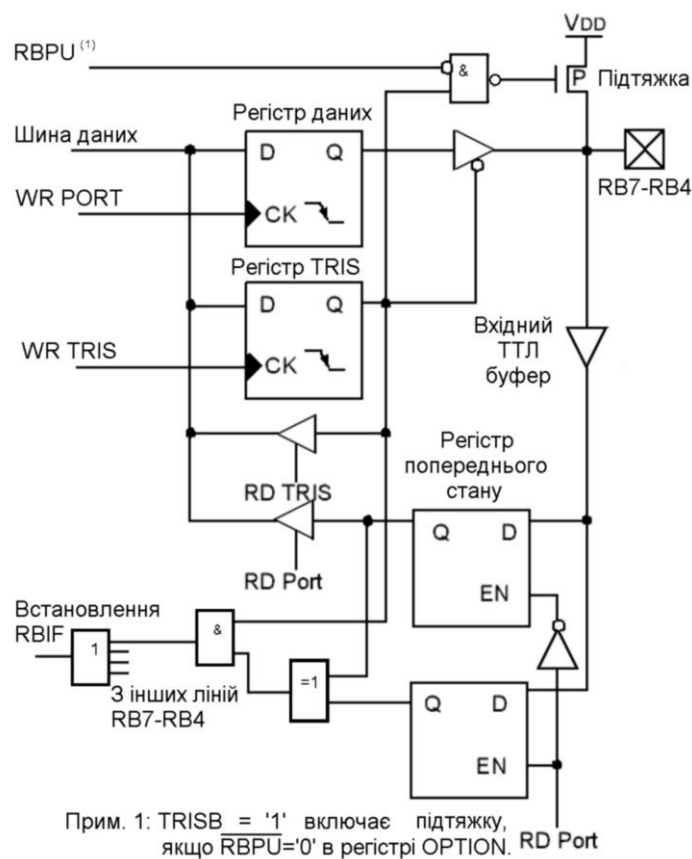


Рис. 91. Схема ліній RB<7:4> порту В

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

установлювані внутрішні активні навантаження на цих чотирьох лініях можуть забезпечити простий інтерфейс, наприклад, із клавіатурою, з виходом із режиму SLEEP після натискання клавіш. Схеми ліній порту В наведено на рис. 91 і рис. 92.

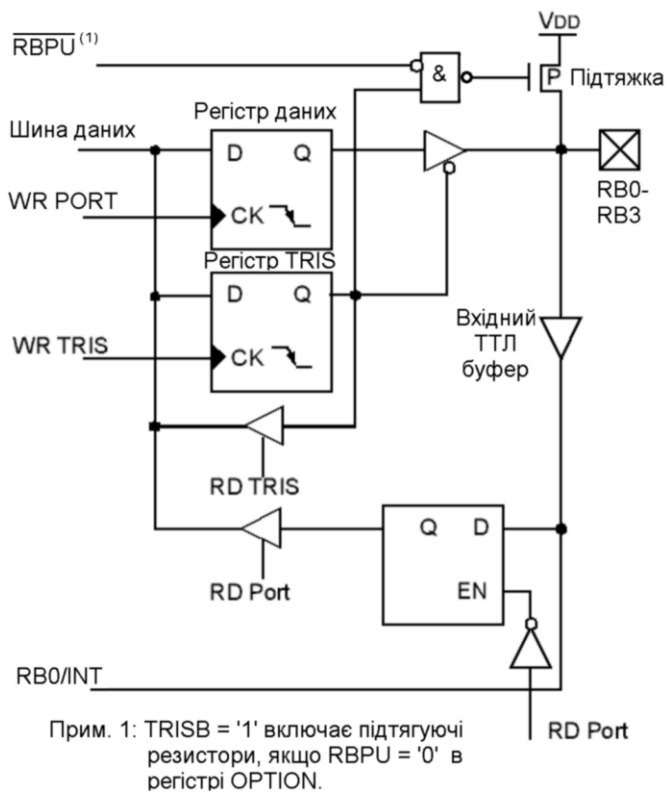


Рис. 92. Схеми ліній RB<3:0> порту В

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. 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 перемикнеться 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:

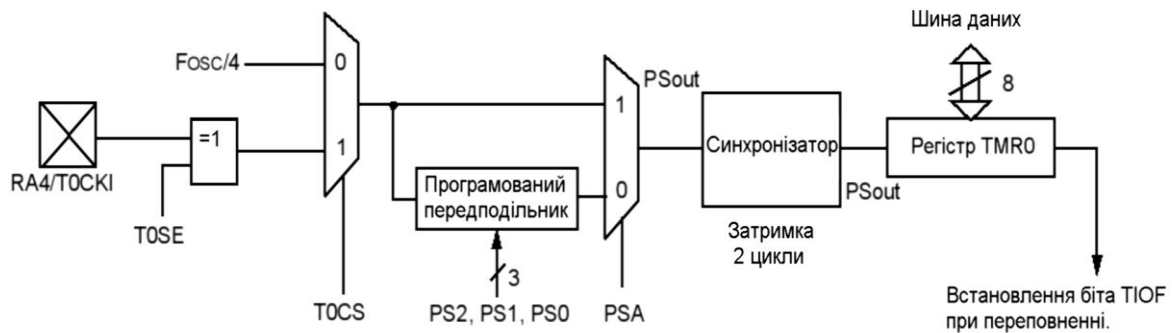
- ◇ 8-розрядний таймер-лічильник `TMR0` з можливістю читання і запису;
- ◇ 8-розрядний програмно керований попередній подільник;
- ◇ мультиплексор для вибору внутрішнього чи зовнішнього тактового сигналу;
- ◇ схему вибору фронту зовнішнього тактового сигналу;
- ◇ формувач запиту переривання після переповнення регістра `TMR0` (зі 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 запису інформації в `TMR0` інкрементування його почнеться після двох командних циклів. Це відбувається з усіма командами, що проводять запис чи модифікацію `TMR0` (наприклад, `MOVF TMR0, CLR F 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/TOCKI`. 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

TMRO, 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

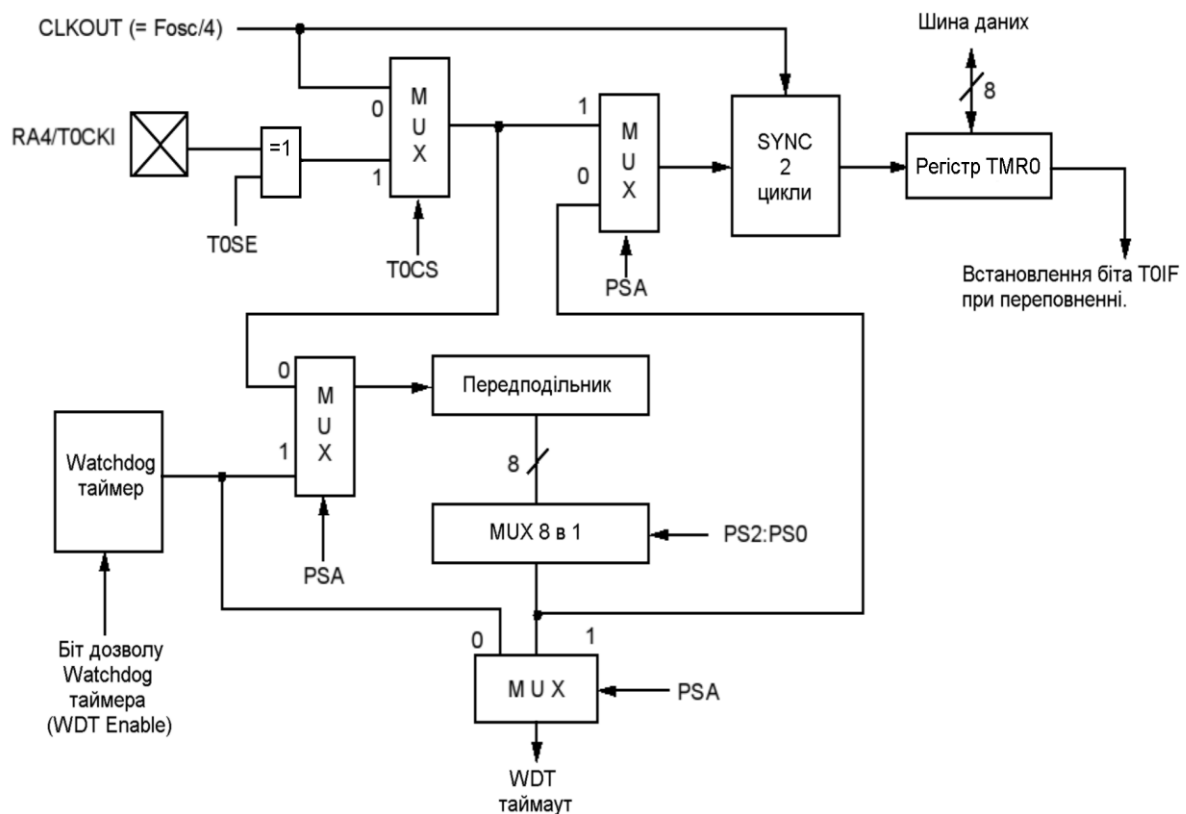


Рис. 94. Структура і можливі варіанти використання подільника

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 Fosc. 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/ТОСКІ були не менше двох періодів тактової частоти Tosc 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/ТОСКІ 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 закінчення 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. Виняток складає переривання в момент завершення запису в EEPROM, ознака якого знаходиться в регістрі EECON1.

Регістр умов переривання (INTCON) є доступним для читання та запису. 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 запису 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 із 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 за переповнення 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 за зміною стану PORTB: нуль вказує, що на жодному з входів RB7-RB4 стан не змінився зі 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 бітів 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 біта забороняє кожне переривання окремо.

Біт дозволу всіх переривань GIE скидається автоматично за таких обставин:

- ◇ за ввімкненням живлення;
- ◇ за зовнішнім сигналом /MCLR при нормальній роботі;
- ◇ за зовнішнім сигналом /MCLR у режимі SLEEP;
- ◇ за закінченням затримки таймера WDT при нормальній роботі;
- ◇ за закінченням 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 В, 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 підпрограмі 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.

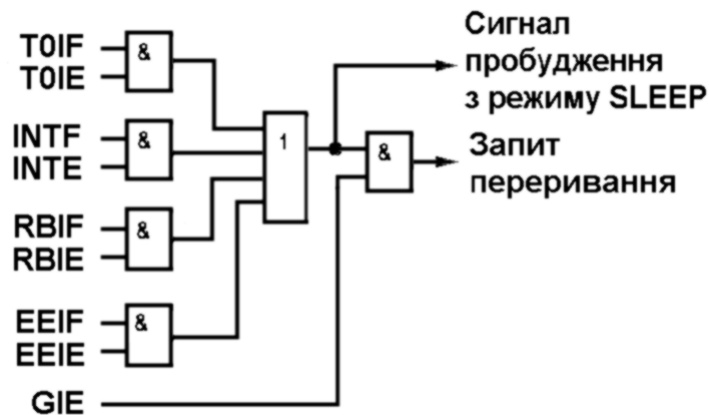


Рис. 96. Логіка переривань мікроконтролера

Зовнішнє переривання на лінії RB0/INT здійснюється або за наростаючим (якщо в регістрі OPTION біт INTEDG=1), або за спадаючим (якщо INTEDG=0) фронтом. 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 зі стану FFH у 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 запису даних в

EEPROM, EEIF (EECON1<4>) встановлюється в одиницю в момент закінчення запису 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

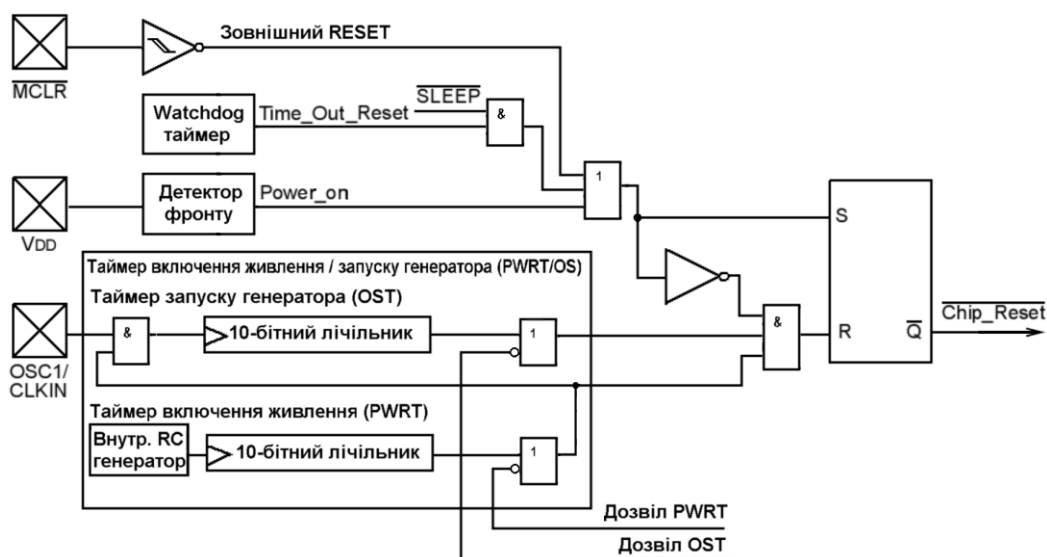


Рис. 97. Схема формування сигналу скидання

PIC16X8X має кілька джерел сигналу скидання:

- ◇ при ввімкненні живлення (Power-on Reset) POR;
- ◇ за зовнішнім сигналом MCLR;
- ◇ за сигналом 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 дочекатися стабілізації напруги живлення в процесі запуску контролера.

Таймер запуску генератора дозволяє після ввімкнення живлення або при виході із режиму зниженого енергоспоживання (SLEEP) дочекатися 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 із зовнішнім формувачем сигналу скидання, наприклад, у системах із 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 вмикає 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 запустити 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 за сигналом MCLR (рис. 98). В інших випадках необхідності в такому колі немає.

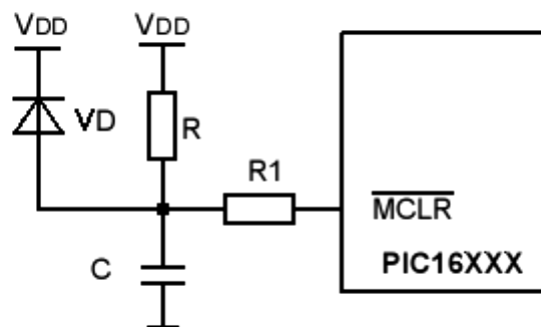


Рис. 98. Зовнішнє коло скидання

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 потрібен 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 якою, 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 запису 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 вимкнене 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, які 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

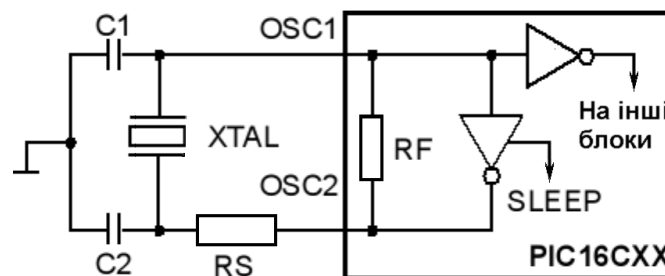


Рис. 99. Під'єднання кварцового резонатора

Схема ввімкнення 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 – близько 33 пФ, для резонаторів на 32 кГц – 68 пФ. Резистор RS може бути потрібний деяким типам 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.

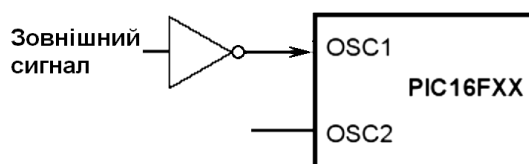


Рис. 100. Синхронізація від зовнішніх джерел

5.15.4. RC-генератор

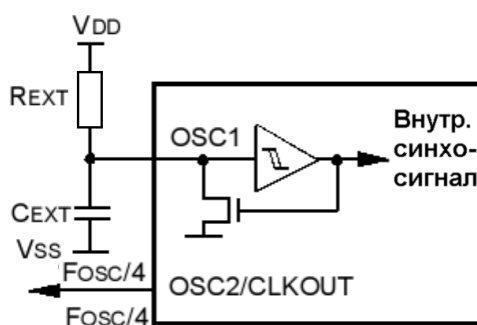


Рис. 101. RC – генератор

Коли не пред'являються високі вимоги до точності відліку часу, зручно для тактування мікросхеми використовувати RC-генератор. Схема ввімкнення 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.

Типові значення $R_{ext} = 5 - 100 \text{ кОм}$, $C_{ext} \geq 20 \text{ пФ}$. 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 й 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

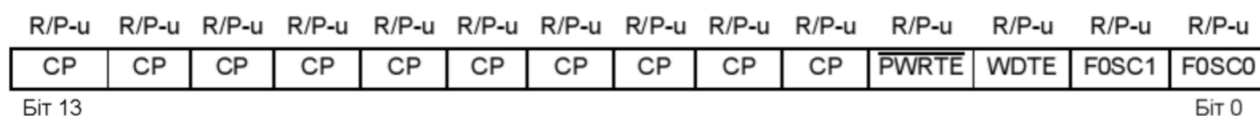


Рис. 102. Формат слова конфігурації

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 бітів слова конфігурації

Позначення	Призначення
CP	Біт захисту коду. Одиниця вказує, що захист коду ввімкнений, нуль – на відсутність захисту
PWRTE	Біт дозволу витримки часу після ввімкнення живлення. Одиниця вказує на наявність витримки в 72 мс при ввімкненні, 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 вказують 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 коду 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	Цикл и	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 I константи і W	1	11 1001 kkkk kkkk	Z	–
ANDWF	Логічне I 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
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

Продовження табл. 46

1	2	3	4	5	6
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 дорівнювати 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 дорівнювати 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 регістра OPTION	1	00 0000 0110 0010	–	–
RETFIE	Повернення з переривання	2	00 0000 0000 1001	–	–
RETURN	Повернення з підпрограми	2	00 0000 0000 1000	–	–

1	2	3	4	5	6
RETLW	Повернення з підпрограми з завантаженням константи у W	2	11 01xx kkkk kkkk	–	–
RLF	Зсув F вліво через ознаку переносу	1	00 1101 dfff ffff	C	1, 2
RRF	Зсув F вправо через ознаку переносу	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 регістра 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 одиничному 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.

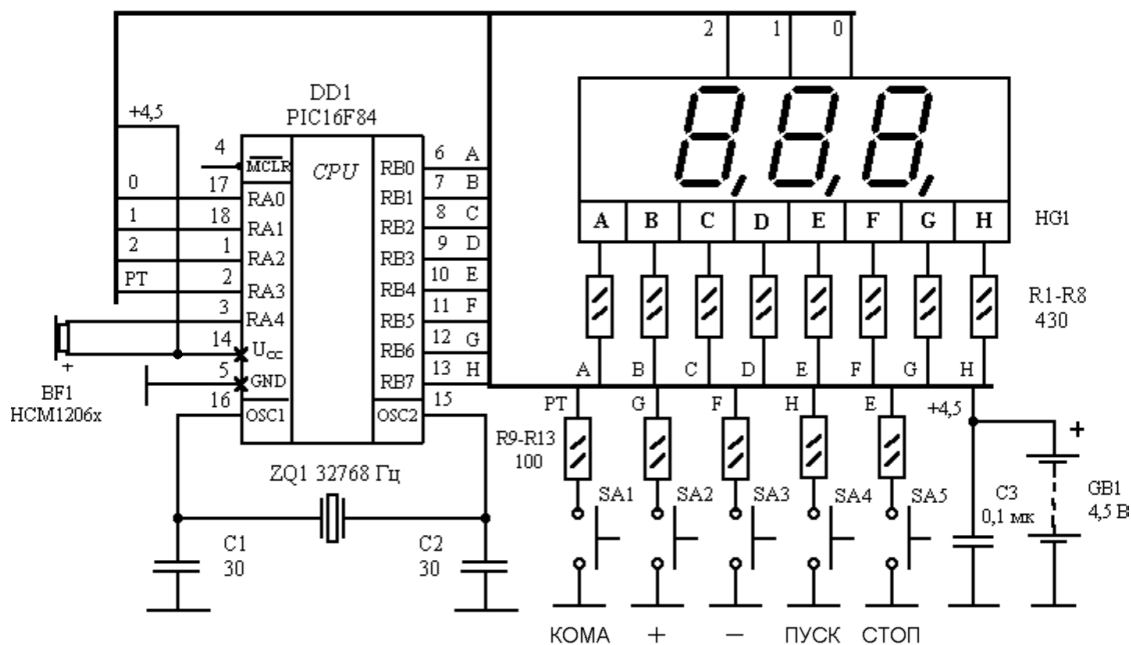


Рис. 103. Універсальний таймер

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 наведено 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 й 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	Ввімкнення циклічного режиму роботи

Виводи RA0-RA3 використовуються як виходи анодів світлодіодів.

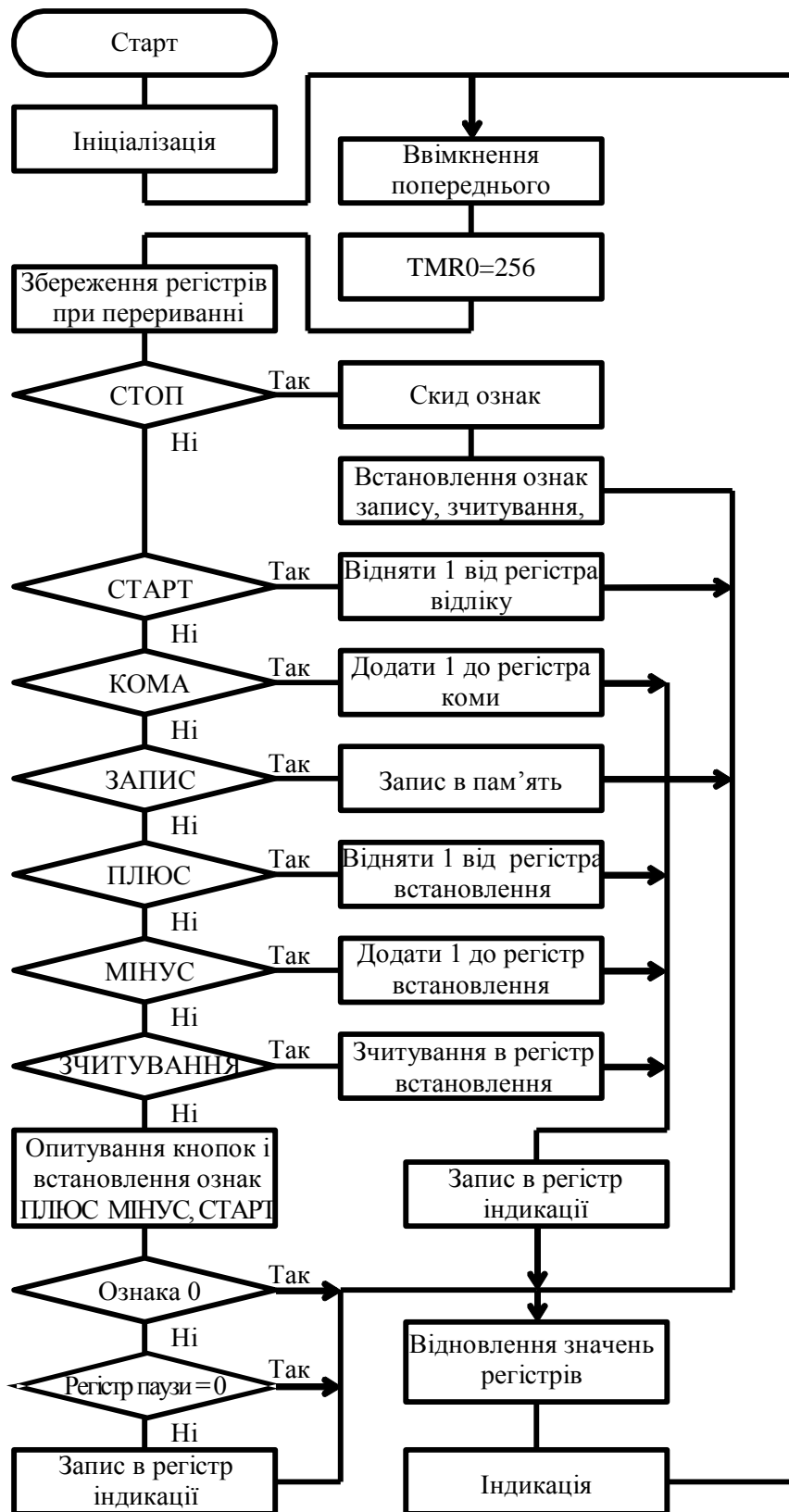


Рис. 104. Алгоритм роботи

Вивід RA4 на схемі навантажений звуковим динаміком HCM1206x із 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 роботи таймера зображений на рис. 104. Після пуску і початкового встановлення реєстрів вмикається таймер із попереднім подільником, які мають такий коефіцієнт поділу, що таймер переповняються 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 з організації динамічної індикації.

Введення значення часового інтервалу здійснюється по одному знаку за допомогою кнопок ПЛЮС та МІНУС. Перехід до наступного знака – за допомогою кнопки КОМА. Ввімкнення коми біля знака 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 залежно від положення коми у момент запуску відліку. Індикація безперервного режиму здійснюється чергуванням ввімкнення 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.

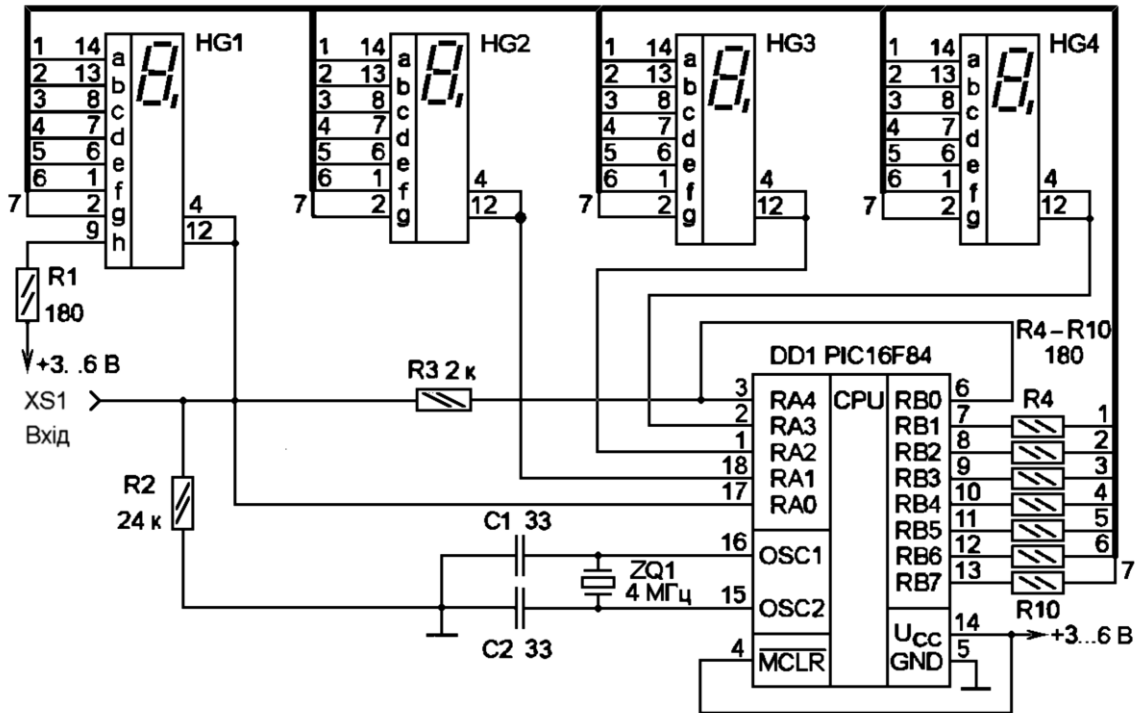


Рис. 105 Схема частотоміра на PIC-контролері

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 під'єднати 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 й 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.

ТЕМА №6. МІКРОКОНТРОЛЕРИ АТМЕГА32

6.1. Характеристики ядра МК AVR

Основними рисами ядра мікроконтролера (МК) АТМega32 є:

- ◇ статична архітектура з мінімальною тактовою частотою, що дорівнює 0;
- ◇ 32 робочі регістри загального призначення;
- ◇ 131 виконувана команда, більшість команд виконується за 1 такт завдяки гарвардській RISC-архітектурі, при цьому досягається продуктивність 1 MIPS на МГц, а максимальна тактова частота може досягати 16 МГц;
- ◇ багаторівнева система переривань, підтримка вкладених переривань;
- ◇ наявність стеку, розміщеного в ОЗП, та можливість операцій з ним;
- ◇ підтримка інтерфейсу JTAG (стандарт IEEE 1149.1) для програмування та відлагодження.

Пам'ять мікро-ЕОМ містить 32 Кб самопрограмованої флеш-пам'яті програм з 104 циклами перепрограмування та з можливістю виділити блок самопрограмування. Для довготривалого зберігання даних використовуються 1024 байти EEPROM, що підтримують 105 циклів перепрограмування з часом зберігання даних протягом 100 років при 25°C.

Для короткочасного зберігання даних використовують 2 Кб внутрішньої пам'яті даних.

Мікроконтролер має також кілька таймерів-лічильників, а саме:

- ◇ два 8-бітні таймери-лічильники з роздільними попередніми подільниками й режимами порівняння, один з лічильників може працювати як лічильник реального часу з окремим генератором;
- ◇ один 16-бітний таймер-лічильник з окремим попереднім подільником, режимами порівняння та фіксацією даних;
- ◇ програмований сторожовий таймер з окремим генератором.

Для вводу та виводу інформації у паралельному коді використовуються:

- ◇ 32 лінії вводу-виводу з програмним конфігуруванням та з програмуванням на вхід або на вихід кожного окремо. Усі лінії мають вхідний буфер з тригером Шмідта. Навантажувальна здатність виходів – 20 мА;
- ◇ чотири канали широтно-імпульсної модуляції;
- ◇ 8-канальний 10-бітний АЦП з можливістю роботи у якості 7-ми канального диференціального АЦП, два диференціальні канали якого мають програмовані коефіцієнти підсилення x1, x10, та x200;
- ◇ аналоговий компаратор.

Для обміну у послідовному коді використовуються:

- ◇ двопровідний послідовний інтерфейс (аналог I²C);
- ◇ стандартний програмований послідовний інтерфейс (USART);
- ◇ інтерфейс SPI з підтримкою веденої та ведучої мікросхем;
- ◇ інтерфейс відлагодження обладнання JTAG.

Мікроконтролер має також систему зовнішніх та внутрішніх джерел переривань, а також шість режимів зниження енергоспоживання.

6.2. Розміщення та призначення виводів МК АТМega32

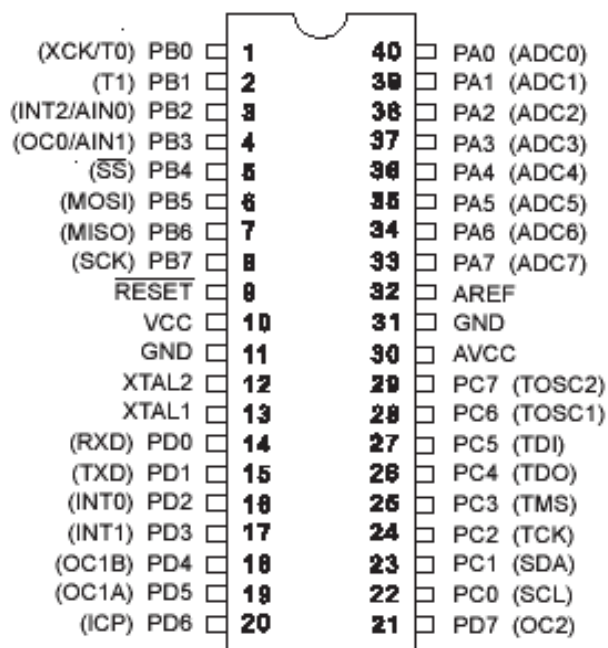


Рис. 106. Розміщення виводів мікросхеми АТМega32

Розміщення виводів наведено на рис. 106, а призначення виводів мікросхеми наведено у таблиці 49.

Таблиця 49

Призначення виводів мікросхеми АТМega32

Позначення	Номер виводу	Тип виводу	Призначення
1	2	3	4
XTAL1	13	Вхід	Вхід інвертора генератора і вхід зовнішнього тактового сигналу
XTAL2	12	Вихід	Вихід інвертора генератора
RESET	9	Вхід	Вхід скидання. При утриманні низького рівня протягом 50 нс відбувається скидання МК
PA0-PA7	40-33	Вхід-вихід	Порт А – 8-розрядний двонаправлений порт вводу-виводу з можливістю аналогового введення даних. До ліній порту можуть під'єднуватися внутрішні навантажувальні резистори (для кожного біту окремо). Вихідні буфери порту симетричні і можуть як віддавати, так і приймати струм. Після скидання лінії порту переводяться у від'єднаний стан. Порт А має можливість вводу аналогових даних
PB0-PB7	1-8	Вхід-вихід	Порт В – 8-розрядний двонаправлений порт вводу-виводу з внутрішніми підтягуючими резисторами, які вмикаються окремо для кожного біту. Вихідні буфери порту симетричні і можуть як віддавати, так і приймати струм. Після скидання лінії порту переводяться у від'єднаний стан. Лінії порту В мають альтернативне призначення, яке вмикається при активації відповідного блоку мікросхеми
PB7/SCK	8	Вхід	Сигнал синхронізації шини SPI
PB6/MISO	7	Вхід-вихід	Сигнал передавання даних для веденої мікросхеми та прийому для ведучої по шині SPI
PB5/MOSI	6	Вхід-вихід	Сигнал передавання даних для ведучої мікросхеми та прийому для веденої по шині SPI
PB4/SS	5	Вхід	Сигнал вибору веденої мікросхеми по шині SPI
PB3/AIN1/OC0	4	Вхід/вихід	Негативний вхід аналогового компаратора. Вихід пристрою порівняння таймера-лічильника 0
PB2/AIN0/INT2	3	Вхід	Позитивний вхід аналогового компаратора. Вхід другого зовнішнього переривання
PB1/T1	2	Вхід	Зовнішній вхід тактового сигналу для таймера-лічильника 1

Продовження таблиці 49

1	2	3	4
PB0/T0/ XCK	1	Вхід	Зовнішній вхід тактового сигналу для таймера-лічильника 0. Сигнал зовнішньої синхронізації послідовного порту (USART)
PC0-PC7	22-29	Вхід-вихід	Порт C – 8-розрядний двонаправлений порт вводу/виводу з внутрішніми навантаженими резисторами, які вмикаються окремо для кожного біту. Вихідні буфери порту симетричні і можуть як віддавати, так і приймати струм. Після скидання лінії порту переводяться у від'єднаний стан. Лінії порту C мають альтернативне призначення, яке вмикається при активації відповідного блоку мікросхеми. Якщо інтерфейс JTAG активований, лінії інтерфейсу активізуються назалежно від сигналу скидання
PC7 TOSC2	29	Вхід	Вхід під'єднання резонатора для таймера-лічильника 2
PC6 TOSC1	28	Вхід	Вхід під'єднання резонатора для таймера-лічильника 2
PC5 TDI	27	Вхід	Вхід даних для інтерфейсу JTAG
PC4 TDO	26	Вихід	Вихід даних для інтерфейсу JTAG
PC3 TMS	25	Вхід	Сигнал вибору тестового режиму інтерфейсу JTAG
PC2 TCK	24	Вхід	Тактовий сигнал інтерфейсу JTAG
PC1 SDA	23	Вхід-вихід	Лінія даних інтерфейсу I2C
PC0 SCL	22	Вхід-вихід	Лінія тактового сигналу інтерфейсу I2C
PD0-PD7	14-21	Вхід-вихід	Порт D – 8-розрядний двонаправлений порт вводу-виводу з внутрішніми навантаженими резисторами, які вмикаються окремо для кожного біту. Вихідні буфери порту симетричні і можуть як віддавати, так і приймати струм. Після скидання лінії порту переводяться у від'єднаний стан. Лінії порту D мають альтернативне призначення, яке вмикається при активації відповідного блоку мікросхеми
PD7 OC2	21	Вихід	Вихід порівняння таймера-лічильника 2
PD6 ICP1	20	Вхід	Вхід фіксації відліку таймера-лічильника 1
PD5 OC1A	19	Вихід	Вихід порівняння А таймера-лічильника 1
PD4 OC1B	18	Вихід	Вихід порівняння В таймера-лічильника 1
PD3 INT1	17	Вхід	Вхід зовнішнього переривання 1
PD2 INT0	16	Вхід	Вхід зовнішнього переривання 0

1	2	3	4
PD1 TXD	15	Вихід	Вихід послідоного порту (USART)
PD0 RXD	14	Вхід	Вхід послідоного порту (USART)
ARef	32	Вхід-вихід	Вхід зовнішнього та вихід внутрішнього опорного сигналу для вбудованого АЦП
GND	11, 31	Живлення	Загальний
Vcc	20	Живлення	Вивід джерела живлення
AVcc	20	Живлення	Вивід джерела живлення аналогових ланцюгів та порту А

6.3. Архітектура ATmega32

Архітектура ATmega32 (рис. 107) є типовою для всього сімейства AVR і виконана за удосконаленою RISC (enhanced RISC) архітектурою. Контролер побудований за гарвардською архітектурою, що дозволяє здійснювати конвеєризацію: під час виконання поточної команди здійснюється вибірка з пам'яті й дешифрування коду наступної команди.

Центральний процесор мікроконтролера складається із лічильника команд, регістра і дешифратора команд, 32-х регістрів загального призначення (РЗП) та арифметико-логічного пристрою (АЛП).

Арифметично-логічний пристрій (АЛП), який виконує всі обчислення, під'єднаний до 32 робочих регістрів, об'єднаних у регістровий файл. Регістри із регістрового файлу майже рівноправні й можуть виконувати функції акумулятора мікропроцесорів i8080.

Зауважимо, що більшість операцій АЛП пристрій виконує за один такт. Регістровий файл та АЛП під'єднані до внутрішньої шини даних.

14-розрядний лічильник команд (PC – Program Counter) використовується для відліку номера комірки пам'яті програм, що містить код команди, яка виконується. Адреса з лічильника через виділену шину надходить на постійну пам'ять ПЗП (Flash). Напрям з програми регістр PC недоступний, при нормальному виконанні програми PC автоматично збільшується на 1 або на 2 залежно від команди, що виконується. Цей порядок порушується при

виконанні команд переходу, виклику й повернення з підпрограм, а також при виникненні переривань. Після увімкнення живлення, а також після скидання МК, в РС автоматично завантажується значення 0x000.

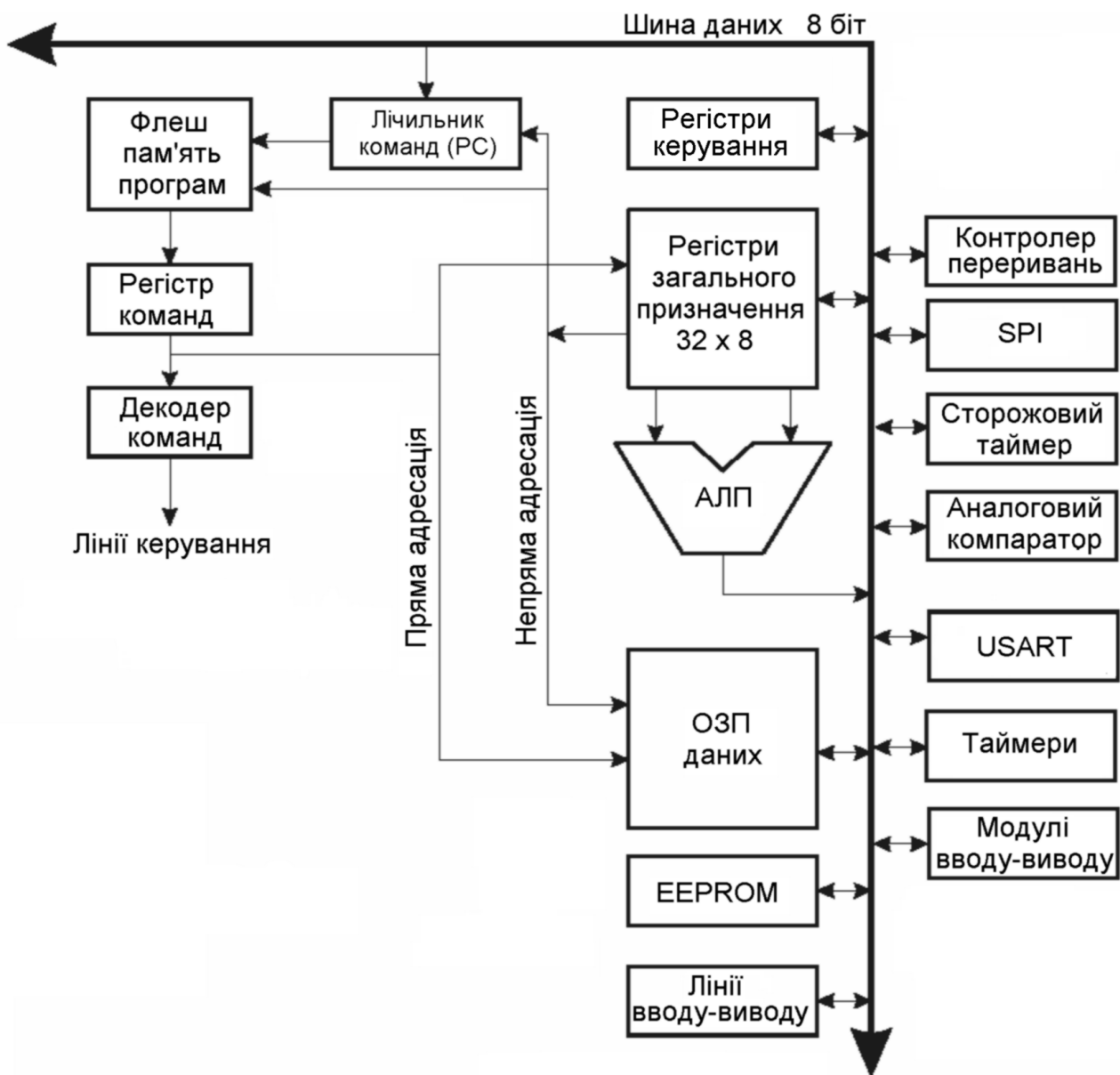


Рис. 107. Архітектура АТМega32

В ПЗП зберігається код команд, які виконуються мікроконтролером. Флеш-пам'ять організована як 16x16 і має об'єм 32 Кб.

В МК AVR майже всі команди (за винятком команд, у яких одним із операндів є 16-розрядна адреса) займають одну комірку пам'яті програм.

Програмування пам'яті здійснюється за допомогою SPI-інтерфейсу безпосередньо на цільовій платі, а також із використанням паралельного

програматора. Кількість циклів перезапису флеш-пам'яті не менша 10^4 . Мікросхема має можливість самопрограмування, тобто мікроконтролер здатний самостійно, без зовнішнього програматора, змінювати вміст комірок пам'яті за програмою, записаною в boot-блоку пам'яті. Це дає можливість змінювати робочу програму з використанням будь-якого інтерфейсу.

Регістр команд та дешифратор команд виконують функції зберігання та дешифрування коду команди, що виконується. Код команди перетворюється в сигнали керування іншими блоками мікроконтролера.

Оперативна пам'ять (ОЗП) використовується для збереження даних, які змінюються в процесі роботи мікроконтролера. ОЗП є у всіх AVR сімейства Tiny, Mega та Classic, крім A90S1200 і її аналогів.

До деяких мікроконтролерів можна під'єднати зовнішню пам'ять даних ємністю до 64 Кб, проте ATmega32 не входить до їх числа.

В ATmega32 об'єм ОЗП становить 2048 байти.

Усі мікроконтролери AVR мають також блок енергонезалежної пам'яті даних з електричним стиранням (EEPROM). Цей тип пам'яті використовують для зберігання даних, різних констант, таблиць перекодувань, каліброваних коефіцієнтів тощо. Дані в EEPROM можна завантажити як через SPI-інтерфейс, так і за допомогою звичайного програматора. В усіх мікроконтролерах EEPROM є доступною для зчитування та побайтової модифікації. Кількість циклів перезапису становить не менше 10^5 .

До периферійних пристроїв мікроконтролера відносять:

- ◇ реєстри керування,
- ◇ таймери,
- ◇ аналогові пристрої,
- ◇ УАПП (UART),
- ◇ сторожовий таймер,
- ◇ порти вводу-виводу,
- ◇ модуль переривань.

Регістри керування призначені для керування роботою мікроконтролера. До них відносять регістр стану SREG, регістр управління MCUCR, вказівник стеку SP тощо. Залежно від типу мікроконтролера склад і кількість регістрів керування може змінюватися.

Кількість таймерів у різних мікросхемах сімейств може бути різною. У ATmega32 до групи таймерів відносять три таймери-лічильники.

Восьмирозрядний таймер T0 використовується для відліку і вимірювання часових інтервалів або як лічильник зовнішніх подій. Регістр відліку таймера може бути записаний і зчитаний. При переповненні таймер генерує запит на переривання.

Таймер T1 (16-розрядний) може генерувати запит на переривання не тільки при переповненні регістра відліку, але й при настанні ряду інших подій.

Усі три таймери можуть працювати в режимі широтно-імпульсного модулятора (ШІМ).

Таймер T1 має можливість фіксації стану таймера за зовнішнім сигналом. Джерелом сигналу для трьох таймерів може бути тактова частота мікроконтролера, поділена на певний коефіцієнт, а також входи T0, T1 та TOSC1 відповідно. Джерела сигналу синхронізації встановлюються незалежно один від одного.

Таймер T2 може працювати як із загальним тактовим сигналом мікросхеми, так і асинхронно відносно інших блоків, використовуючи власний тактовий генератор.

До складу аналогових пристроїв в ATmega32 входить:

- аналоговий компаратор та
- 8-канальний 10-розрядний АЦП.

Аналоговий компаратор може порівнювати значення сигналу з позитивного входу AIN0 з сигналом на негативному вході AIN1. Вихід аналогового компаратора може бути налаштований на роботу на вхід захоплення таймера T1, або формувати власне переривання.

ATMega32 у своєму складі має прийомопередавач (UART). Швидкість передавання даних може змінюватися в широких межах. Модуль UART може виявляти і сигналізувати про різні збої при передаванні даних, а саме: переповнення, помилка кадрування, невірний стоп-біт. Для зменшення ймовірності збоїв у модулі реалізована функція фільтрації завад. Для взаємодії з програмою в модулі передбачено три переривання за наступними подіями: ”передавання даних завершено“, ”регістр даних передавача порожній“, ”прийом завершено“.

Виводи МК, які використовуються модулем UART, є лініями порту D. В якості входу приймача (RxD) використовується вивід PD0, а в якості виходу передавача (TxD) – вивід PD1.

Сторожовий таймер WDT (WATCHDOG) призначений для перезапуску програми у випадку появи збою у ході її виконання. Програма, що працює без збоїв, періодично скидає сторожовий таймер, не допускаючи його переповнення. Сторожовий таймер має свій власний RC - генератор, який працює на частоті 1 МГц. На вході WDT увімкнено попередній подільник входної частоти з програмованим коефіцієнтом ділення, що дозволяє регулювати часовий інтервал переповнення таймера і скидання мікроконтролера.

Мікросхема має 32-і незалежні лінії вводу-виводу, згруповані в 4 порти. Кожна з ліній може запрограмуватися на ввід або на вивід. Потужні вихідні драйвери забезпечують струмову навантажувальну спроможність 20 мА на лінію, при цьому загальне струмове навантаження на всі лінії одного порту не має перевищувати 80 мА. Зауважимо, що інші мікроконтролери AVR мають кількість ліній вводу-виводу, що коливається від 3 до 53.

Модуль переривань містить схему дозволу/заборони та ранжування запитів за пріоритетом. У ATMega32 переривання діляться на внутрішні та зовнішні. Джерелами внутрішніх переривань є вбудовані модулі (наприклад, таймер T0). Зовнішні переривання викликаються скиданням (сигналом на виводі RESET) або сигналами на виводах INT. В МК AVR усім перериванням

поставлений у відповідність власний вектор переривання – адреса в початковій області пам'яті програм, за якою розташовується команда переходу до підпрограми обробки переривання.

Внутрішній тактовий генератор AVR-МК можна запускати від декількох джерел опорної частоти (зовнішній генератор, зовнішній кварцовий резонатор, внутрішня або зовнішня RC-ланка). Оскільки AVR-МК повністю статичні, мінімальну допустиму частоту нічим не обмежено, тобто можна легко забезпечити навіть покроковий режим виконання програми. Максимальна робоча частота визначається конкретним типом мікроконтролера.

6.4. Функціонування конвеєра, цикл виконання команд мікроконтролера AVR

У МК процес виконання команд організований так, щоб при виборі команди з пам'яті програм відбувалося виконання попередньої команди, тобто функціонує дворівневий конвеєр. Таким чином, тривалість машинного циклу дорівнює тривалості періоду тактової частоти. Робота цього конвеєра наведена на рис. 108.

Під час першого машинного циклу відбувається вибірка команди з пам'яті програм і її декодування.

Під час другого циклу ця команда виконується, а паралельно відбувається вибірка і декодування другої команди. В результаті фактичний час виконання кожної команди дорівнює одному машинному циклу.

При виконанні певних команд може відбуватися порушення нормальної роботи конвеєра. Типовим прикладом таких команд є команди умовного переходу. Якщо умова, яка перевіряється командою умовного переходу, істинна, то виконання програми буде продовжено з нової адреси. Оскільки в конвеєрі вже відбулася вибірка команди, розташованої після команди переходу, то час виконання команди переходу збільшується на 1 цикл, під час якого відбувається вибірка команди, розташованої за потрібною адресою.

Аналогічно команди безумовного відносного і непрямого переходу,

команди виклику підпрограм, команди повернення з підпрограм також змінюють вміст РС. У результаті виконання цих команд відбувається розрив у роботі конвеєра, а внаслідок цього – затримання виконання програми на 2...4 машинні цикли.

За тієї ж причини відбувається порушення роботи конвеєра при виникненні переривання. Мінімальна затримка при цьому становить 4 машинні цикли.

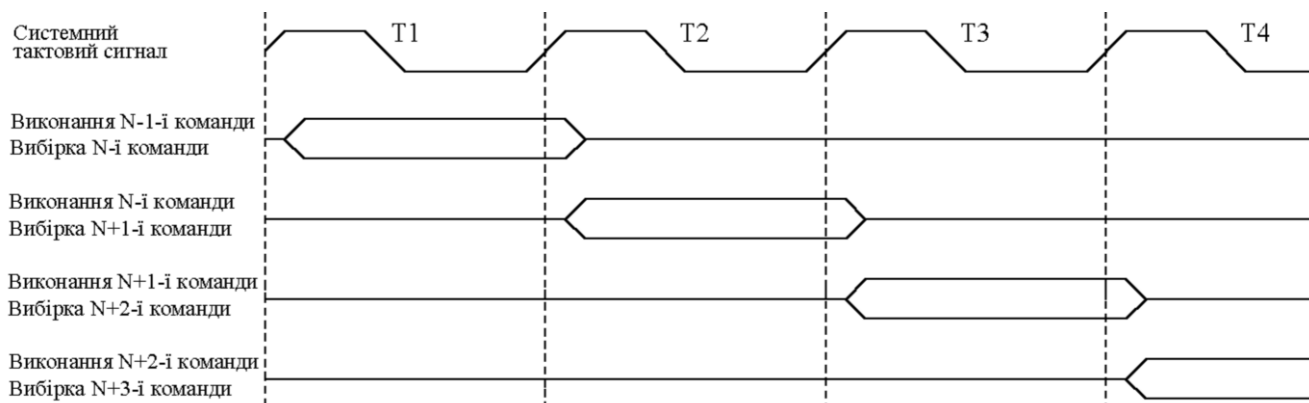


Рис. 108. Робота конвеєра

6.5. Регістр стану - SREG

Регістр стану містить інформацію про результати виконання останньої арифметичної або логічної команди. Ця інформація може використовуватися для того, щоб змінити процес виконання програми, в командах умовного переходу та інших умовних операторах.

Регістр стану мікроконтролера SREG має такий формат (рис. 109):

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 109. Формат регістра SREG

Біт 7 — I: загальний дозвіл переривань. Внаслідок встановлення цієї ознаки в одиничний стан визначається робота всієї системи переривань. Окремі види переривань вмикаються і вимикаються за допомогою додаткових регістрів конфігурації.

Якщо ознака "Загальний дозвіл переривань" має нульове значення, всі переривання заблоковані, незалежно від того, увімкнені вони чи ні в додаткових регістрах конфігурації. Ознака I апаратно скидається відразу після виклику відповідної процедури обробки переривання і встановлюється при виконанні команди RETI, дозволяючи подальші переривання. Ознака I може бути також встановлена і скинута програмно за допомогою команд SEI і CLI відповідно.

Біт 6 — T: ознака користувача – біт для тимчасового зберігання інформації. Біт T використовується командами BLD (завантаження біту T) і BST (читання біту T) як комірок для тимчасового зберігання інформації. Будь-який біт будь-якого регістра загального призначення може копіюватися в T, а потім вміст T може, в свою чергу, копіюватися в будь-який інший біт того ж або будь-якого іншого регістра.

Біт 5 — H: ознака половинного перенесення. Ця ознака встановлюється в одиницю, якщо мало місце перенесення з молодшої половини байту (з 3-го розряду в 4-й) або зворотне перенесення із старшої половини байту при виконанні деяких арифметичних операцій. Цей біт є аналогічним ознаці AC мікропроцесорів Intel.

Біт 4 — S: ознака знака, $S = N \otimes V$. Ця ознака є результатом операції "Виключне АБО" (XOR) між ознаками N (від'ємний результат) і V (переповнювання числа в додатковому коді). Відповідно, ця ознака встановлюється в одиницю, якщо результат виконання арифметичної операції менший нуля.

Біт 3 — V: ознака переповнення додаткового коду. Ця ознака використовується при роботі зі знаковими числами (числами, представленими в додатковому коді). Ознака встановлюється в одиницю, якщо в результаті арифметичної операції відбудеться переповнення числа, представленого в додатковому коді.

Біт 2 — N: ознака від'ємного значення. Ознака встановлюється в

одиницю, якщо в результаті арифметичної операції старший розряд результату дорівнює одиниці. Якщо старший розряд результату дорівнює нулю, то ознака N теж дорівнює нулю.

Біт 1 — Z: ознака нуля. Ознака встановлюється в одиницю, якщо результат виконаної операції дорівнює нулю.

Біт 0 — C: ознака перенесення. Ця ознака відображає переповнення результату (перенесення в старший розряд) при виконанні арифметичної операції. Крім того, ознака перенесення використовується в операціях зсувів.

6.6. Організація пам'яті даних ATmega32

Пам'ять МК AVR виконана за гарвардською архітектурою з розділеними адресними просторами пам'яті програм і пам'яті даних.

Пам'ять програм призначена для зберігання команд, що керують роботою МК, а також констант. Пам'ять програм в МК AVR – це Flash-ПЗП з кількістю циклів перезапису не менше 104. Оскільки більшість команд займають у пам'яті 16 бітів, пам'ять програм має 16-розрядну організацію.

Пам'ять даних складається з трьох областей: регістрової пам'яті (регістровий файл), статичного ОЗП і пам'яті EEPROM. Оскільки регістрова пам'ять знаходиться в адресному просторі ОЗП, то про ці дві області кажуть як про одну. Область EEPROM розташована у своєму власному адресному просторі (рис. 110).

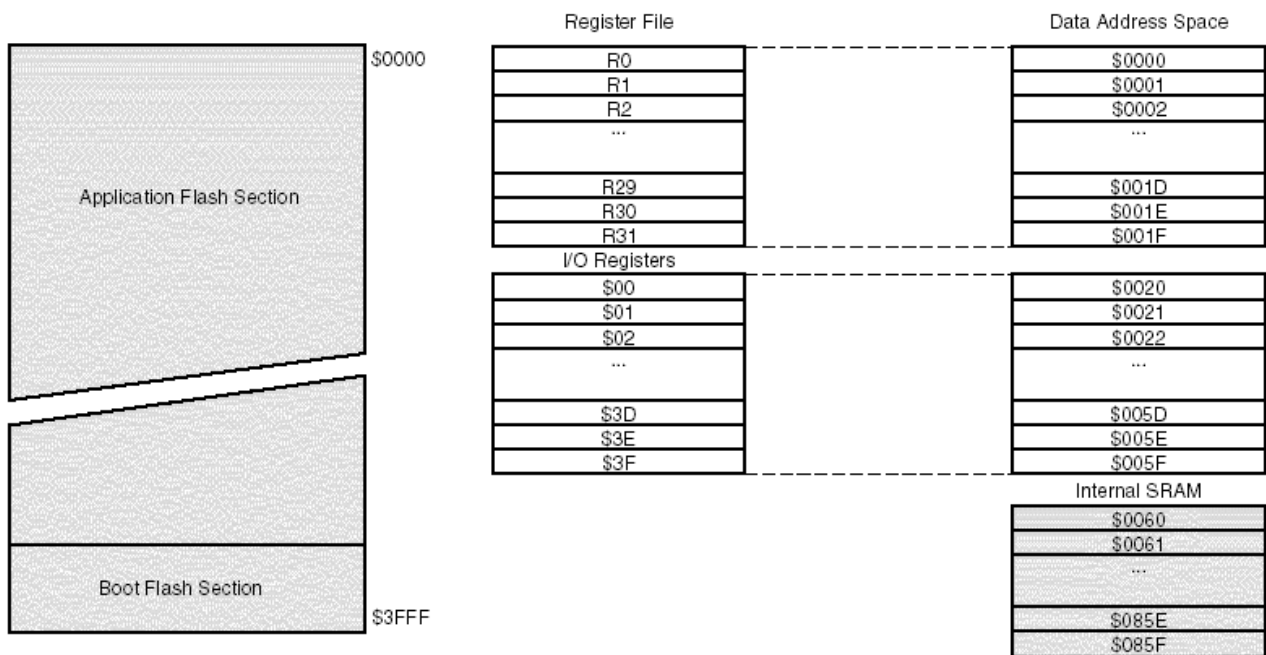


Рис. 110. Структура пам'яті МК АТМega32

6.6.1. Регістровий файл

Регістровий файл займає молодші 32 байти у загальному адресному просторі AVR (рис. 111). Шість із 32 регістрів файлу можна використовувати як три 16-розрядні покажчики адреси у процесі непрямого адресування даних. Один із цих покажчиків (*Z Pointer*) застосовують також для доступу до даних, записаних у пам'яті програм мікроконтролера. Використання трьох 16-розрядних покажчиків (*X*, *Y* і *Z Pointers*) істотно підвищує швидкість пересилання даних під час роботи прикладної програми.

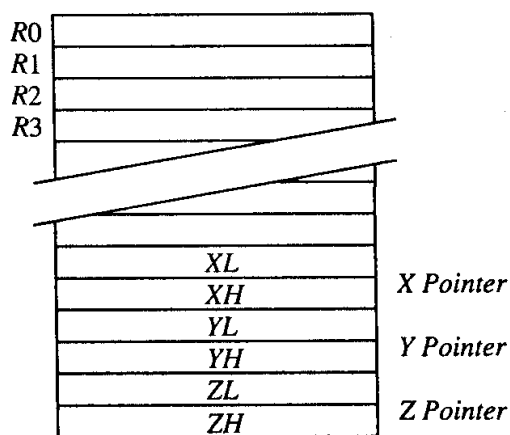


Рис. 111. Структура регістрового файлу

6.6.2. Регістри вводу-виводу

В області регістрів вводу-виводу розташовані різні службові регістри (регістр показчика стеку, регістр стану та ін.), а також регістри керування периферійними пристроями МК.

До регістрів вводу-виводу можна звертатися двома способами: як до регістрів (за допомогою команд IN та OUT) і як до комірок ОЗП. У першому випадку використовуються адреси регістрів вводу-виводу з простору вводу-виводу (0x00...0x3F). У другому випадку адресу регістра вводу-виводу треба збільшити на 0x20.

6.6.3. Стек

В МК ATmega32 стек реалізований у загальному ОЗП і його глибина визначається лише розміром вільної області пам'яті даних. Оскільки область SRAM займає 2 Кб, то розмір показчика стеку (Stack Pointer) становить 11 біт. На відміну від деяких інших мікроконтролерів, у ATmega32 стек росте у бік зменшення адреси. В якості показчика стеку використовуються два регістри вводу-виводу (рис. 112): SPL, розташований за адресою регістра вводу-виводу 0x3D (0x5D - ОЗП), та SPH, розташований за адресою вводу-виводу 0x3E (0x5E - ОЗП).

Вказівник стеку повністю доступний з програми. Крім того, в системі команд МК є команди занесення в стек (PUSH) і добування зі стеку (POP). Після включення живлення або після скидання показчик стеку дорівнює нулю, тому на самому початку програми його необхідно проініціалізувати, записавши в нього значення верхньої адреси області пам'яті даних, призначеної для стека.

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Рис. 112. Вказівник стеку

Наприклад:

```
RESET:    rjmp MAIN
          .
          .
MAIN:     ldi R16, 0x00 ; Завантажити регістр r16
          ldi R17, 0x01 ; Завантажити регістр r17
          out SPL, R16  ; Ініціалізація вказівника
          out SPH, R17  ; стеку
```

Під час виклику підпрограми адреса наступної команди зберігається в стеку. Значення покажчика стеку зменшується на 2, оскільки для зберігання лічильника команд потрібно 2 байти. При поверненні з підпрограми ця адреса витягується зі стеку і завантажується у лічильник команд (PC), а потім значення покажчика стеку збільшується на 2. Те ж саме відбувається під час переривання.

6.6.4. Пам'ять EEPROM

ATMega32 містить 1024 байти пам'яті EEPROM даних. Вона організована як окремий простір даних, в якій кожен байт може бути прочитаний і записаний. EEPROM має витривалість щонайменше 105 циклів запису та стирання. Доступ до EEPROM здійснюється через регістр адреси EEADR, регістр даних EEData та регістр керування. Ці регістри доступні в просторі вводу/виводу.

Час доступу при запису даних у EEPROM становить приблизно 8,5 мс (8448 циклів каліброваного внутрішнього RC генератора). Сама пам'ять може повідомити програмі про можливість запису наступного байту. З метою запобігання ненавмисних записів у EEPROM необхідно дотримуватися відповідної процедури запису.

Коли EEPROM читається, процесор зупиняється на чотири такти до наступної виконаної команди. Коли в EEPROM здійснюється запис, процесор зупиняється на два такти до виконання наступної команди.

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	EEAR9	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

Рис. 113. Формат регістра адреси EEPROM

Формат регістра адреси EEPROM показаний на рис. 113. Старші біти регістра зарезервовані, використовуються лише біти EEAR9..0, що задають адресу комірки EEPROM, яка читається чи записується.

Формат регістра даних – на рис. 114. Через цей регістр здійснюється читання та запис даних EEPROM. Для керування роботою використовується регістр EECR, зображений на рис. 115.

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 114. Регістр даних EEPROM

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

Рис. 115. Формат регістра EECR

Біти регістра EECR мають таке призначення:

EERIE – біт дозволу переривань. Дозволяє переривання від EEPROM, якщо біт встановлений в 1. Переривання генеруються у момент закінчення запису.

EEMWE – біт "ключа" для запису в EEPROM. Коли EEMWE встановлено, встановлення біту EEWE починає запис у EEPROM за вибраною адресою. Якщо EEMWE дорівнює нулю, встановлення EEWE не починає запис. Біт EEMWE автоматично скидається у 0 через 4 цикли, тому команда встановлення біту EEWE має йти відразу за командою встановлення біту

EEMWE. При запису одиничного значення біту EEMWE не слід встановлювати біт EEWE.

EEWE – біт початку запису у EEPROM. Сигнал починає запис у випадку встановлення в 1 біту EEMWE. Біт автоматично скидається в 0 у момент закінчення запису. Програма може опитувати цей біт і чекати появи нуля перед новим звертанням до пам'яті.

EERE – біт дозволу читання. Коли в регістр EEAR записана вірна адреса, біт EERE має бути встановлений в 1 для читання даних. Читання EEPROM займає одну команду і тому дані будуть доступними негайно. Біт автоматично скидається в 0.

Слід дотримуватися такої процедури при запису в EEPROM (кроки 3 та 4 не обов'язкові):

1. Зачекати, поки EEWE стане дорівнювати нулю.
2. Зачекати, поки біт SPEN в SPMCR стане дорівнювати нулю.
3. Задати нову адресу EEPROM для EEAR.
4. Задати нові дані EEPROM для EEDR.
5. Встановити біт EEMWE в 1, а біти EEWE та EECR – у нуль.
6. Протягом чотирьох тактів після встановлення EEMWE записати логічну 1 в біт EEWE.

EEPROM не може бути запрограмована у процесі запису флеш-пам'яті програм. Програма повинна перевірити, що програмування флеш-пам'яті програм завершено до початку нового циклу запису в EEPROM. Крок 2 потрібний лише, якщо програмне забезпечення містить Boot Loader. Якщо флеш-пам'ять програм не оновлюється програмою, крок 2 може бути опущений. Переривання між кроками 5 та 6 і зробить цикл запису невдалим, тому що біт EEMWE буде скинутий у нуль. Тому в процесі програмування EEPROM переривання мають бути заборонені.

6.7. Система команд мікроконтролера AVR

Усі команди мікроконтролерів AVR можна поділити на:

- ◇ - команди пересилання даних;
- ◇ - команди арифметичних операцій та порівняння;
- ◇ - команди операцій із бітами;
- ◇ - команди керування системою;
- ◇ - команди безумовного передавання даних;
- ◇ - команди умовного передавання даних, керування та пропуску команд.

В описі команд використовуються певні позначення, які наведено у наступній таблиці.

Таблиця 50

Позначення, що застосовуються в описі команд

Позначення	Регістри й операнди
Rd	Регістр призначення (і джерело) в реєстровому файлі
Rr	Регістр джерело в реєстровому файлі
R	Результат виконання команди
K	Константа (8 біт)
k	Дані адреси константи для лічильника програм
b	Біт у реєстровому файлі або I/O реєстр (3 біти)
s	Біт у реєстрі статусу (3 біти)
X, Y, Z	Регістр непрямої адресації (X=R27:R26, Y=R29:R28, Z=R31:R30)
P	Адреса порту вводу-виводу
q	Зсув при прямій адресації (6 біт)
STACK	Стек для адреси повернення й опущених у стек реєстрів
SP	Показчик стеку

6.7.1. Група команд пересилання даних

Команди пересилання даних призначені для пересилання даних між реєстрами, між реєстрами та комірками пам'яті даних та програм, а також для пересилання даних у порти вводу-виводу. Команди не змінюють реєстр ознак. Усі команди пересилання даних наведено у наступній таблиці.

Таблиця 51

Група команд пересилання даних

Мнемоніка	Опис	Операція	Цикли
1	2	3	4
MOV Rd, Rr	Пересилання між регістрами загального призначення	Rd <- Rr	1
MOVW Rd, Rr	Пересилання між парами регістрів	Rd+1:Rd<- Rr+1:Rr	1
LDI Rd, K	Завантаження константи в регістр	Rd<-K	1
LD Rd, X	Непряме читання	Rd <- [X]	2
LD Rd, X+	Непряме читання з постінкрементом	Rd<-[X], X<- X+1	2
LD Rd, -X	Непряме читання з переддекрементом	X<-X-1, Rd<-[X]	2
LD Rd, Y	Непряме читання	Rd <- [Y]	2
LD Rd, Y+	Непряме читання з постінкрементом	Rd<-[Y], Y<- Y+1	2
LD Rd, -Y	Непряме читання з переддекрементом	Y<Y-1, Rd<-[Y]	2
LDD Rd, Y+q	Непряме відносне читання	Rd <- [Y+q]	2
LD Rd, Z	Непряме читання	Rd <- [Z]	2
LD Rd, Z+	Непряме читання з постінкрементом	Rd<-[Z], Z<-Z+1	2
LD Rd, -Z	Непряме читання з переддекрементом	Z<-Z-1, Rd<-[Z]	2
LDD Rd, Z+q	Непряме відносне читання	Rd <- [Z+q]	2
LDS Rd, k	Читання з ОЗП даних	Rd <- [k]	2
ST X, Rr	Непрямий запис	[X] <- Rr	2
ST X+, Rr	Непрямий запис із постінкрементом	[X]<- Rr, X<- X+1	2
ST -X, Rr	Непрямий запис переддекрементом	X<-X-1, [X] <- Rr	2
ST Y, Rr	Непрямий запис	[Y]<- Rr	2
ST Y+, Rr	Непрямий запис з постінкрементом	[Y]<-Rr, Y<- Y+t	2
ST -Y, Rr	Непрямий запис з переддекрементом	Y<-Y-1, [Y]<-Rr	2
STD Y+q, Rr	Непрямий відносний запис	[Y+q] <- Rr	2
ST Z, Rr	Непрямий запис	[Z] <- Rr	2
ST Z+, Rr	Непрямий запис з постінкрементом	[Z] <-Rr, Z<- Z+1	2
ST-Z, Rr	Непрямий запис з переддекрементом	Z<-Z-1, [Z]<-Rr	2

Закінчення таблиці 51

1	2	3	4
STD Z+q, Rr	Непрямий відносний запис	$[Z+q] \leftarrow Rr$	2
STS k, Rr	Запис в ОЗП	$[k] \leftarrow Rr$	2
LPM	Завантаження даних з пам'яті програм у регістр R0	$R0 \leftarrow \{Z\}$	
LPM Rd, Z	Завантаження даних з пам'яті програм	$Rd \leftarrow \{Z\}$	3
LPM Rd, Z+	Завантаження даних з пам'яті програм і постікремент Z	$Rd \leftarrow (Z),$ $Z \leftarrow Z+1$	3
SPM	Запис у програмну пам'ять	$\{Z\} \leftarrow R1:R0$	-
IN Rd, P	Пересилання з порта у регістр	$Rd \leftarrow P$	1
OUT P, Rr	Пересилання з регістра у порт	$P \leftarrow Rr$	1
PUSH Rr	Збереження байта в стеку	$STACK \leftarrow Rr$	2
POP Rd	Витягання байта зі стека	$Rd \leftarrow STACK$	2

6.7.2. Група команд арифметичних операцій та порівняння

До групи арифметичних операцій відносять команди, які записані у наступній таблиці. Як бачимо з таблиці, арифметичні команди можуть використовувати для додавання і віднімання, збільшення та зменшення, множення як цілих так і дробових чисел. Команда ділення відсутня. Результат виконання команди може бути записаний у довільний регістр.

Таблиця 52

Група команд арифметичних операцій

Мнемоніка	Опис	Операція	Цикли	Ознаки
1	2	3	4	5
ADD Rd, Rr	Додавання двох регістрів	$Rd \leftarrow Rd + Rr$	1	Z, C, N, V, H
ADC Rd, Rr	Додавання двох регістрів з бітом перенесення	$Rd \leftarrow Rd + Rr + C$	1	Z, C, N, V, H
ADIW Rd, K	Додавання регістрової пари з константою	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	2	Z, C, N, V, S
SUB Rd, Rr	Віднімання двох регістрів	$Rd \leftarrow Rd - Rr$	1	Z, C, N, V, H
SUBI Rd, K	Віднімання константи із регістра	$Rd \leftarrow Rd - K$	1	Z, C, N, V, H
SBC Rd, Rr	Віднімання двох регістрів із урахуванням біта перенесення	$Rd \leftarrow Rd - Rr - C$	1	Z, C, N, V, H
SBCI Rd, Ko	Віднімання константи із урахуванням біта перенесення	$Rd \leftarrow Rd - K - C$	1	Z, C, N, V, H

Закінчення таблиці 52

1	2	3	4	5
SBIW Rd, K	Віднімання константи із реєстрової пари	Rdh:Rdl <- Rdh: Rdl - K	2	Z, C, N, V, S
DEC Rd	Декремент реєстра	Rd <- Rd - 1	1	Z, N, V
INC Rd	Інкремент реєстра	Rd <- Rd + 1	1	Z, N, V
MUL Rd,Rr	Беззнакове множення	R1:R0 <- Rd x Rr	2	Z, C
MULS Rd,Rr	Множення чисел зі знаком	R1:R0 <- Rd x Rr	2	Z, C
MULSU Rd,Rr	Множення числа зі знаком з числом без знака	R1:R0 <- Rd x Rr	2	Z, C
FMUL Rd, Rr	Дробове множення чисел без знака	R1:R0 <- Rd x Rr<<1	2	Z, C
FMULS Rd, Rr	Дробове множення чисел зі знаком	R1:R0 <- Rd x Rr<<1	2	Z, C
FMULSU Rd, Rr	Дробове множення числа зі знаком з числом без знака	R1:R0 <- Rd x Rr<<1	2	Z, C

Команди порівняння призначені для аналізу вмісту реєстра. Команди дуже подібні за властивостями до різних модифікацій команди віднімання, але результат операції порівняння нікуди не записується, модифікуються лише біти реєстра ознак. Звичайно, команди порівняння використовуються разом із командами умовного переходу.

Таблиця 53

Група команд порівняння

Мнемоніка	Опис	Операція	Цикли	Ознаки
CP Rd, Rr	Порівняння двох реєстрів	Rd - Rr	1	Z, N, V, C, H
CPC Rd, Rr	Порівняння двох реєстрів із урахуванням перенесення	Rd - Rr - C	1	Z, N, V, C, H
CPI Rd, K	Порівняння реєстра з константою	Rd - K	1	Z, N, V, C, H

6.7.3. Група команд роботи з бітами

До групи відносять команди логічних операцій. Ці команди записані у наступній таблиці.

Таблиця 54

Група команд роботи з бітами

Мнемоніка	Опис	Операція	Цикли	Ознаки
1	2	3	4	5
AND Rd, Rr	Логічне І двох регістрів	$Rd \leftarrow Rd \cdot Rr$	1	Z, N, V
ANDI Rd, Ko	Логічне І регістра та константи	$Rd \leftarrow Rd \cdot K$	1	Z, N, V
EOR Rd, Rr	Виключаюче або двох регістрів	$Rd \leftarrow Rd \oplus Rr$	1	Z, N, V
OR Rd, Rr	Логічне або двох регістрів	$Rd \leftarrow Rd \vee Rr$	1	Z, N, V
ORI Rd, K	Логічне або регістра та константи	$Rd \leftarrow Rd \vee K$	1	Z, N, V
CLR Rd	Скидання всіх розрядів регістра	$Rd \leftarrow 0$	1	Z, N, V
SER Rd	Установка всіх розрядів регістра	$Rd \leftarrow 0FFH$	1	-
TST Rd	Перевірка регістра на від'ємне та нульове значення	$Rd \leftarrow Rd \cdot Rd$	1	Z, N, V
NEG Rd	Перетворення у додатковий код	$Rd \leftarrow 00H - Rd$	1	Z, C, N, V, H
COM Rd	Перетворення у зворотний код (інверсія регістра)	$Rd \leftarrow 0FFH - Rd$	1	Z, C, N, V

До групи команд операцій з розрядами відносять команди, які записані у наступній таблиці.

Таблиця 55

Група команд операцій з розрядами

Мнемоніка	Опис	Операція	Цикли	Ознаки
1	2	3	4	5
CBR Rd, K	Скид розрядів регістра	$Rd \leftarrow Rd \cdot \overline{K}$	1	Z, N, V
SBR Rd, K	Установка розряду чи розрядів регістра	$Rd \leftarrow Rd \vee K$	1	Z, N, V
CBIA, b	Скид розряду порта вводу-виводу	$A.b \leftarrow 0$	2	-
SBI A, b	Установка розряду вводу-виводу	$A.b \leftarrow 1$	2	-
BCLRs	Скид ознаки	$SREG.s \leftarrow 0$	1	SREG.s
BSETs	Установка ознаки	$SREG.s \leftarrow 1$	1	SREG.s
BLD Rd, b	Завантаження розряду регістра з біта T (SREG)	$Rd, b \leftarrow T$	1	-
BST Rr, b	Запис розряду регістра загального призначення в біт T (SREG)	$T \leftarrow Rd.b$	1	T
CLC	Скид біта перенесення	$C \leftarrow 0$	1	C

1	2	3	4	5
SEC	Установка біта перенесення	C<- 1	1	C
CLN	Скид біта від'ємного числа	N<-0	1	N
SEN	Установка біта від'ємного числа	N<- 1	1	N
CLZ	Скид біта нуля	Z<-0	1	Z
SEZ	Установка біта нуля	Z<-1	1	Z
CLI	Загальна заборона переривань	I<- 0	1	I
SEI	Загальний дозвіл переривань	I <- 1	1	I
CLS	Скид біта знака	S<-0	1	S
SES	Установка біта знака	S<-1	1	S
CLV	Скид біта переповнення для додаткового кода	V<-0	1	V
SEV	Установка біта переповнення додаткового кода	V <-1	1	V
CLT	Скид біта Т користувача	T<-0	1	T
SET	Установка біта Т користувача	T<- 1	1	T
CLH	Скид біта половинного перенесення	H<-0	1	H
SEH	Установка біта половинного перенесення	H<-1	1	H

До групи команд зсувів відносять команди зсуву вправо чи вліво через біт перенесення та без нього, а також команді обміну місцями тетрами.

Таблиця 56

Група команд зсувів

Мнемоніка	Опис	Операція	Цикли	Ознаки
ASR Rd	Арифметичний зсув вправо	Rd7 -> Rd6 -> Rd5 -> Rd4 -> Rd3 -> Rd2 -> Rd1 -> Rd0	1	Z, C, N, V
LSL Rd	Логічний зсув вліво	C <- Rd7 <- Rd6 <- Rd5 <- Rd4 <- Rd3 <- Rd2 <- Rd1 <- Rd0 <- 0	1	Z, C, N, V
LSR Rd	Логічний зсув вправо	0 -> Rd7 -> Rd6 -> Rd5 -> Rd4 -> Rd3 -> Rd2 -> Rd1 -> Rd0 -> C	1	Z, C, N, V
ROLRd	Зсув вліво через перенесення	C <- Rd7 <- Rd6 <- Rd5 <- Rd4 <- Rd3 <- Rd2 <- Rd1 <- Rd0 <- C	1	Z, C, N, V
ROR Rd	Зсув вправо через перенесення	C -> Rd7 -> Rd6 -> Rd5 -> Rd4 -> Rd3 -> Rd2 -> Rd1 -> Rd0 -> C	1	Z, C, N, V
SWAP Rd	Обмін місцями тетрами	Rd(3—0) <-> Hd(7—4)	1	-

Група команд керування мікросхемою

Таблиця 57

Група команд керування мікросхемою

Мнемоніка	Опис	Цикли
NOP	Нема операції	1
SLEEP	Перехід у "сплячий" режим	3
WDR	Скид сторожового таймера	1
BREAK	Зупинка програми. Команда використовується лише при відлагодженні програми	-

Команди групи призначені для задавання режиму роботи мікросхеми та виконують інші допоміжні дії. Команди групи не змінюють регістра ознак.

Команди передавання даних керування

Група команд передавання даних керування призначена для зміни послідовності виконання команд, організації циклів тощо. Команди такого типу поділяються на команди безумовних переходів, що здійснюють перехід незалежно від жодної умови, команди умовних переходів, які перевіряють умови і здійснюють переходи залежно від умов та команди пропуску за умовою. Команди останнього типу пропускають наступну умову, якщо умова виконалась.

До команд безумовних переходів також віднесені команди виклику та повернення з підпрограм та переривань. Усі команди безумовних переходів наведено у наступній таблиці.

Таблиця 58

Команди безумовних переходів

Мнемоніка	Опис	Операція	Цикли
RJMP k	Відносний безумовний перехід	$PC \leftarrow PC + k + 1$	2
IJMP	Непрямий безумовний перехід	$PC \leftarrow Z$	2
JMP k	Прямий безумовний перехід	$PC \leftarrow k$	3
RCALL k	Відносний виклик підпрограми	$STACK \leftarrow PC + 1$ $PC \leftarrow PC + k + 1$	3

Закінчення таблиці 58

ICALL	Непрямий виклик підпрограми	STACK <- PC+1 PC<-Z	3
CALL k	Прямий виклик підпрограми	STACK <- PC+1 PC<-k	4
RET	Повернення з підпрограми	PC <- STACK	4
RETI	Повернення з підпрограми обробки переривань	PC <- STACK I = 1	4

Таблиця 59

Пропуску команди за умовою

Мнемоніка	Опис	Цикли
CPSE Rd, Rr	Порівняння і пропуск наступної команди при рівності (Rd = Rr)	1/2/3
SBRC Rr, b	Пропуск наступної команди, якщо біт регістра скинутий (Br.b = 0)	1/2/3
SBRS Rr, b	Пропуск наступної команди, якщо біт регістра встановлений (Rr.b = 1)	1/2/3
SBIC A, b	Пропуск наступної команди, якщо біт регістра скинутий (A.b = 1)	1/2/3
SBIS A, b	Пропуск наступної команди, якщо біт регістра встановлений (A.b = 1)	1/2/3

Група команд пропуску команди за умовою дозволяє виконати прості дії залежно від умови без створення розгалуження програми. Всі команди цієї групи можуть дозволити пропускати наступну команду за різних умов. Команди пропуску команди за умовою наведено у наступній таблиці.

Команди умовних переходів здійснюють перехід, якщо певний біт чи біти регістра ознак встановлені у задані значення. Якщо ж біти не встановлені, то здійснюється перехід до наступної команди. Всі команди цієї групи виконують перехід за відносною адресою.

Таблиця 60

Група команд умовних переходів

Мнемоніка	Опис	Цикли
BRBC s, k	Перехід, якщо ознака S регістра SREG скинутий (результат останньої операції додатний)	1/2
BRBS S, k	Перехід, якщо ознака S регістра SREG встановлений (результат останньої операції від'ємний)	1/2
BRCS k	Перехід, є перенесення, тобто біт C = 1	1/2
BRCC k	Перехід, нема перенесення, тобто біт C = 0	1/2
BREQ k	Перехід, якщо результат нульовий, (Z = 1)	1/2
BRNE k	Перехід, якщо результат не нульовий (Z = 0)	1/2
BRSH k	Перехід за умовою "більше або рівно", тобто коли C = 0	1/2
BRLO k	Перехід за умовою "менше", тобто коли C = 1	1/2
BRMI k	Перехід за умовою "від'ємне значення" для чисел без знака, якщо N = 1	1/2
BRPL k	Перехід за умовою "додатне значення" для чисел без знака, якщо біт N = 0	1/2
BRGE k	Перехід за умовою "більше або рівно" для чисел зі знаком, тобто за умови $(N \oplus V) = 0$	1/2
BRLT k	Перехід за умови "менше" для чисел зі знаком, тобто коли $(N \oplus V) = 1$	1/2
BRHS k	Перехід за половинним перенесенням, коли біт H = 1	1/2
BRHC k	Перехід, якщо немає половинного перенесення, коли біт H = 0	1/2
BRTS k	Перехід, якщо біт користувача T встановлений (T = 1)	1/2
BRTC k	Перехід, якщо біт користувача T скинутий (T = 0)	1/2
BRVS k	Перехід за переповненням у додатковому коді V = 1	1/2
BRVC k	Перехід, якщо немає переповнення у додатковому коді V = 0	1/2
BRID k	Перехід, якщо переривання заборонені (I = 0)	1/2
BRIE k	Перехід, якщо переривання дозволені (I = 1)	1/2

6.8. Порти вводу-виводу

МК ATmega32 має 4 порти вводу-виводу (ПВВ) A ... D, кожен з яких є восьмирозрядним. Отже, загальна кількість ліній вводу/виводу дорівнює 32.

Конфігурування кожної лінії порту (вказання напрямку передавання даних) може проводитися програмно в будь-який момент часу. Вхідні буфери портів побудовані за схемою тригера Шмідта. Для ліній, сконфігурованих як

вхідні, є можливість під'єднання внутрішнього підтягуючого резистора опором 30...120 кОм між входом і живленням.

Максимальна навантажувальна здатність вихідних буферів ПВВ при логічному 0 на виході становить 20 мА.

Звертання до портів відбувається через регістри вводу-виводу, причому під кожен порт в адресному просторі вводу/виводу зарезервовано три адреси. Під цими адресами розташовуються регістри:

- ◇ даних портів PORT_x,
- ◇ напрямку даних DDR_x,
- ◇ виводів порту PIN_x.

При скиданні МК регістри DDR_x і PORT_x очищуються, а всі виводи портів після скидання встановлюються в третій (високоімпедансний) стан. Можна задавати конфігурацію кожного виводу незалежно від решти.

PIN_x насправді не є регістрами, за цими адресами здійснюється доступ до фізичних значень сигналів на виводах порту. Відповідно, вони доступні лише для читання, тоді як PORT_x і DDR_x доступні і для читання, і для запису.

Запис у порт означає запис потрібного стану для кожного виводу порту у відповідний регістр даних порту PORT_x. А читання стану порту виконується або читанням регістра даних порту PORT_x, або регістра виводів порту PIN_x.

На рис. 116 зображена спрощена структурна схема розряду порту.

При читанні регістра виводів порту PIN_x відбувається зчитування логічних рівнів сигналів, присутніх на виводах порту. А при читанні регістра даних порту PORT_x відбувається зчитування даних, які знаходяться в регістрі-засувці порту. Це справедливо як для вхідних, так і для вихідних контактів.

Напрямок передавання даних визначається вмістом регістра передавання даних DDR_x. Якщо розряд DD_{xn} цього регістра встановлений в 1, відповідний n-й вивід порту є виходом. Якщо ж розряд DD_{xn} цього регістру скинутий в 0, відповідний вивід порту є входом.

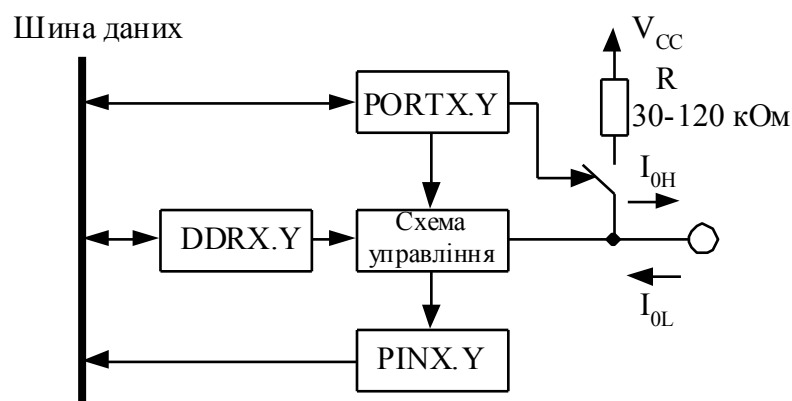


Рис. 116. Розряд порту вводу-виводу

Управління підтягуючим резистором здійснюється за допомогою регістра даних порту PORTx. Якщо розряд Pxn регістра PORTx встановлений в 1 і відповідний вивід порту є входом, між цим виводом і проводом живлення під'єднується підтягуючий резистор. Щоб від'єднати підтягуючий резистор, необхідно або скинути відповідний розряд регістра PORTx, або зробити вивід порту виходом (таблиця 61).

Таблиця 61

Задання режиму роботи порту

DDxn	Pxn	Функція виводу	Резистор	Опис
0	0	Вхід	Від'єднаний	Третій високоімпедансний стан (Hi-Z)
0	1	Вхід	Під'єднаний	При під'єднанні виводу до спільного проводу він є джерелом струму
1	0	Вихід	Від'єднаний	Вихід встановлений в 0
1	1	Вихід	Від'єднаний	Вихід встановлений в 1

6.9. Система переривань

Мікроконтролери ATmega32 мають кілька різних джерел переривань. Усі переривання мають індивідуальні біти дозволу, у які має бути записана 1 для дозволу відповідного переривання. Крім того, є спільний загальний біт дозволу переривань, яким зручно блокувати усі переривання, коли переривання основної програми неможливе.

Коли відбувається переривання, біт глобального дозволу переривань скидається і всі переривання блокуються. У процесі обробки переривань можна

встановити біт глобального дозволу переривань для виконання вкладених переривань. Також біт глобального дозволу переривань автоматично встановлюється в 1 при виконанні команди повернення з переривання RETI.

Таблиця 62

Розміщення векторів скидання та переривань

Номер вектора	Адреса підпрограми	Назва	Визначення
1	\$000+V1	RESET	Скидання від зовнішнього виводу, за ввімкненням чи пропаданням живлення, від Watchdog-таймера або сигналу від JTAG
2	\$002+V2	INT0	Зовнішній запит переривань 0
3	\$004+V2	INT1	Зовнішній запит переривань 1
4	\$006+V2	INT2	Зовнішній запит переривань 2
5	\$008+V2	TIMER2 COMP	Сигнал порівняння таймера-лічильника 2
6	\$00A+V2	TIMER2 OVF	Сигнал порівняння таймера-лічильника 2
7	\$00C+V2	TIMER1 CAPT	Сигнал фіксації таймера-лічильника 1
8	\$00E+V2	TIMER1 COMPA	Сигнал порівняння А таймера-лічильника 1
9	\$010 + V2	TIMER1 COMPB	Сигнал порівняння В таймера-лічильника 1
10	\$012 + V2	TIMER1 OVF	Переповнення таймера-лічильника 1
11	\$014 + V2	TIMER0 COMP	Сигнал порівняння таймера-лічильника 0
12	\$016 + V2	TIMER0 OVF	Сигнал переповнення таймера-лічильника 0
13	\$018 + V2	SPI	Сигнал закінчення передавання даних через послідовний порт SPI
14	\$01A + V2	USART	Сигнал закінчення приймання даних через USART
15	\$01C + V2	USART	Сигнал вільності буфера передавання USART
16	\$01E + V2	USART	Сигнал закінчення передавання даних через USART
17	\$020 + V2	ADC	Перетворення ADC завершено
18	\$022 + V2	EE_RDY	EEPROM готовий
19	\$024 + V2	ANA_CO MP	Аналоговий компаратор

Розміщення векторів скидання та переривань

Номер вектора	Адреса підпрограми	Назва	Визначення
20	\$026 + V2	TWI	Сигнал переривання від двовивідного послідовного інтерфейсу TWI
21	\$028 + V2	SPM_RDY	Пам'ять програм готова до обміну

Є два основні типи переривань.

Перший тип переривань має ознаку, що встановлюється за деякою подією, а скидання ознаки здійснюється у процесі обробки. Якщо переривання не може бути оброблене, ознака збереже свій стан і переривання буде викликане при першій можливості.

У другому типі переривання викликається за станом сигналу і, якщо за деякий час переривання не буде оброблене, то воно буде втрачене.

Зауважимо, що при використанні інструкції CLI переривання буде негайно від'єднане, навіть якщо воно надійшло у процесі виконання інструкції CLI. Після обробки переривання в основній програмі буде виконано хоча б одну команду перед будь-яким новим перериванням. Також варто відзначити, що реєстр статусу не зберігається автоматично при вході в переривання, а його збереження має бути виконане програмою користувача. Для кожного джерела переривань є окремий вектор, розміщений у спеціальній області пам'яті програм. Найнижчі адреси в пам'яті програм за замовчуванням призначені для вектора скидання та векторів переривань. Повний список векторів показано в таблиці нижче.

Параметри V1 та V2 визначаються бітами BOOTRST та IVSEL. Біт BOOTRST задається у слові конфігурування мікросхеми, а біт IVSEL задається у головному реєстрі керування перериваннями.

Якщо біт BOOTRST запрограмовано, пристрій перейде до адреси скидання у boot-блоці. Коли біт IVSEL в реєстрі GICR встановлено, вектори переривань будуть переміщені на початок boot-блоку флеш-пам'яті програм. Усі можливі варіанти значень V1 та V2 наведено в таблиці 63.

Можливі варіанти значень V1 та V2

BOOTRST	IVSEL	V1	V2
1	0	\$0000	\$0000
1	1	\$0000	Початок boot-блоку
0	0	Початок boot-блоку	\$0000
0	1	Початок boot-блоку	Початок boot-блоку

Порядок у списку також визначає пріоритет рівнів переривань. Адреса, що має менший номер, задає вищий рівень пріоритету. Скидання (RESET) має найвищий пріоритет, а наступним іде INTO – зовнішній запит переривання 0. Деякі із векторів переривань можуть переміщуватися у процесі роботи за допомогою встановлення біту IVSEL в головному реєстрі управління перериванням (GICR).

Формат головного реєстра управління перериванням GICR зображено на рис. 117.

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	-	-	-	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 117. Головний реєстр керування перериваннями GICR

Біти INT2 - INTO – біти дозволів зовнішніх запитів переривань. Логічна 1 дозволяє відповідний зовнішній запит переривань.

Біт IVCE використовується у якості "ключа" для зміни стану біта IVSEL. Для зміни стану біта IVSEL необхідно, щоб біт IVCE був встановлений у 1. Біт IVCE скидається через 4 цикли після запису даних або після запису даних у біт IVSEL. Для уникнення переходів на хибні вектори переривань встановлення біта IVCE в одиницю забороняє переривання.

6.10. Лічильники-таймери мікросхеми

Мікросхема ATmega32 у своїй структурі має три таймери-лічильники:

- ◇ 8 - бітний таймер-лічильник 0 (Timer/Counter0);
- ◇ 16 - бітний таймер-лічильник 1 (Timer/Counter1);

◇ 8 - бітний таймер-лічильник 2 (Timer/Counter2).

Кожен із таймерів-лічильників орієнтований на виконання свого кола задач і має відповідні особливості, наприклад, таймер-лічильник 2 має незалежне джерело синхронізації.

Розглянемо будову та роботу кожного із лічильників окремо.

6.10.1. Попередні подільники таймерів-лічильників

Попередній подільник призначений для попереднього поділу вхідної частоти на певний коефіцієнт і використовується для збільшення періоду відліку тактового сигналу без збільшення розрядності таймерів-лічильників.

Таймер-лічильник 1 і таймер-лічильник 0 використовують один і той же модуль попереднього подільника, але можуть мати різні налаштування коефіцієнта попереднього поділу. Таймер-лічильник 2 використовує власний попередній подільник.

Будова попереднього подільника таймерів-лічильників 0 та 1 зображена на рис. 118. Таймери-лічильники 0 та 1 можуть бути синхронізовані безпосередньо системним тактовим сигналом, що забезпечує найбільшу швидкість відліку. Крім того, таймери-лічильники 0 та 1 можуть синхронізуватися тактовим сигналом від одного з чотирьох відводів від попереднього подільника із частотою синхронізації, меншою за частоту тактового сигналу на 8, 64, 256 чи 1024 ($f_{CLK_I/O}/8$, $f_{CLK_I/O}/64$, $f_{CLK_I/O}/256$, або $f_{CLK_I/O}/1024$ відповідно).

На жаль, попередній подільник таймерів-лічильників 0 та 1 не дозволяє ділити сигнал із зовнішнього входу. Вибір потрібного джерела синхронізації здійснюється за допомогою бітів CS12-CS10 для таймера-лічильника 1 та бітами CS02-CS00 для таймера-лічильника 0.

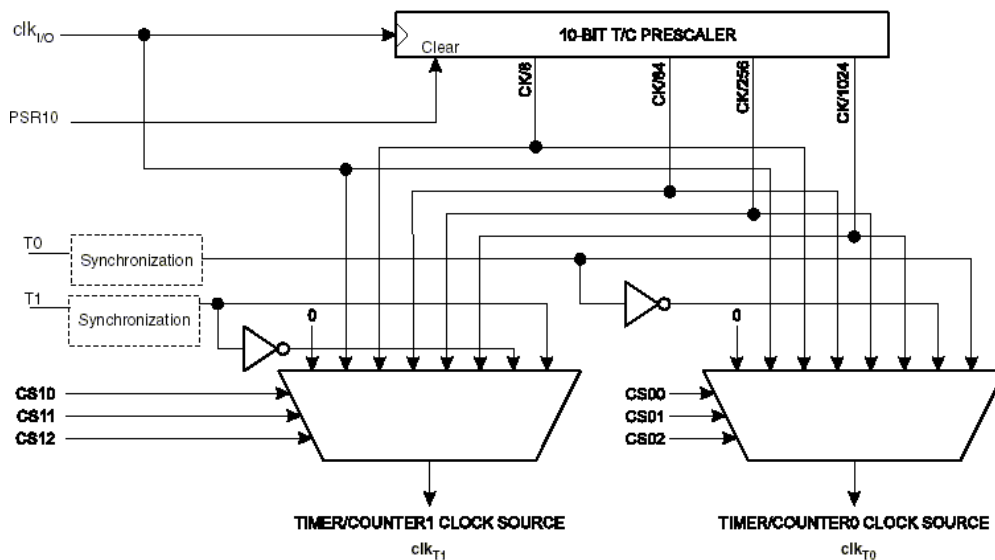


Рис. 118. Будова попереднього подільника таймерів-лічильників 0 та 1

Попередній подільник таймера-лічильника 2 може ділити частоту вхідного сигналу. Будова попереднього подільника таймера-лічильника 2 зображена на рис. 119. Зі структурної схеми попереднього подільника таймера-лічильника 2 бачимо, що він може синхронізуватися тактовим сигналом від одного з шести відводів від попереднього подільника із частотою синхронізації, меншою за частоту вхідного сигналу у 8, 32, 64, 128, 256 чи 1024 разів. Джерело вхідного сигналу попереднього подільника таймера-лічильника 2 обирається бітом AS2 регістра ASSR і дозволяє під'єднати до входу попереднього подільника як зовнішній вхід, так і системний тактовий сигнал.

Вибір потрібного джерела синхронізації здійснюється за допомогою бітів CS22-CS20.

Попередній подільник не має синхронізації від таймерів-лічильників і працює незалежно від них. Наприклад, якщо попередній подільник налаштовано на коефіцієнт поділу 1024, то перший імпульс після встановлення режиму роботи лічильника може надійти і через 1, і через 1023 тактових сигнали. Якщо необхідно встановити час до надходження першого імпульсу, можна скористатися бітами скидання попередніх подільників.

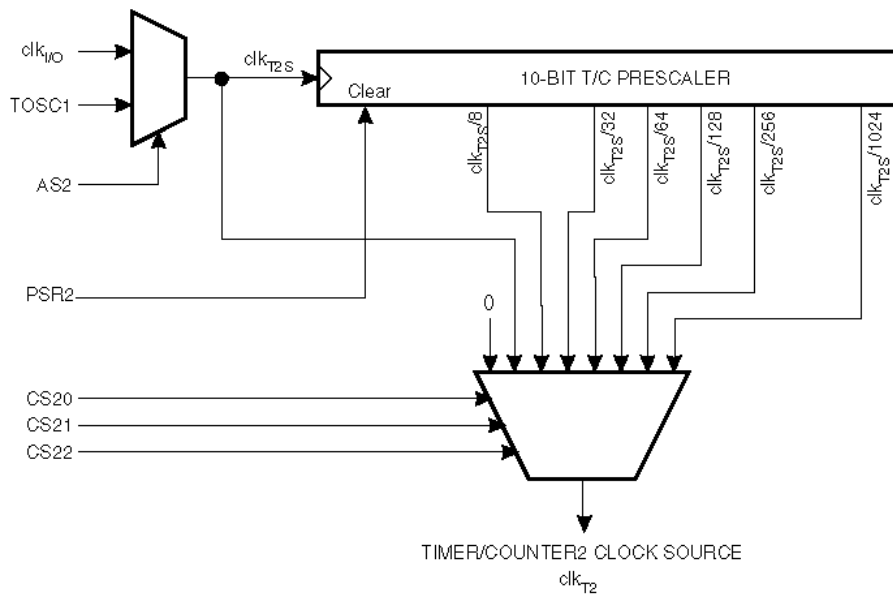


Рис. 119. Будова попереднього подільника таймера-лічильника 2

Bit	7	6	5	4	3	2	1	0	SFOIR
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 120. Регістр SFOIR

Біт скидання попереднього подільника таймерів-лічильників 0 та 1 носить назву PSR10, а біт PSR2 скидає попередній подільник таймера-лічильника 2. Біти PSR10 та PSR2 знаходяться у регістрі SFOIR, формат регістра SFOIR наведено на рис. 120.

Запис у біт PSR10 одиниці скидає попередній подільник таймерів-лічильників 0 та 1, аналогічно запис біту PSR2 скидає попередній подільник таймера-лічильника 2. Самі біти PSR10, PSR2 скидаються апаратно після виконання відповідної операції. Запис у біти PSR10, PSR2 нуля не виконує жодної дії. При читанні бітів PSR10 та PSR2 завжди читається нуль, якщо скидання виконане.

6.10.2. Восьмирозрядні таймери-лічильники 0 та 2

Таймери-лічильники 0 та 2 – це універсальні восьмирозрядні лічильники з модулем порівняння і підтримкою функцій широтно-імпульсної модуляції (ШИМ або PWM). Вони дозволяють формувати задані проміжки часу для роботи в режимі реального часу, а також можуть використовуватися як генератори сигналів.

Восьмирозрядні таймери-лічильники 0 та 2 дуже подібні за своєю будовою, програмуванням та режимами роботи.

Основні особливості таймерів- лічильників 0 та 2:

- ◇ наявність модуля порівняння;
- ◇ подвійна буферизація при запису в регістри порівняння;
- ◇ скидання таймера при рівності значення заданому;
- ◇ наявність симетричного широтно-імпульсного модулятора;
- ◇ два незалежних джерела переривання TOV0, OCF0 для таймера-лічильника 0, а також TOV2, OCF2 для таймера-лічильника 2.

Таймер-лічильник 2 має іншу будову попереднього подільника та може працювати у асинхронному режимі.

Вхідним сигналом таймера-лічильника 0 може бути як зовнішній сигнал з виводу мікросхеми, так і сигнал із тактового генератора, пропущений через попередній подільник. Спрощена блок-схема восьмирозрядного таймера-лічильника 0 наведена на рис. 121.

Підрахунок кількості імпульсів здійснює лічильник TCNT0 таймера-лічильника. Сигнали про напрям відліку, необхідність скидання та імпульси, що підраховуються, надходять на нього із блоку керування (Control Logic), який керується регістром керування TCCR0. Також блок керування (Control Logic) видає сигнал переповнення таймера TOV0.

Імпульси, що підраховуються, надходять на блок керування від блоку вибору сигналу синхронізації (Clock Select), котрий визначає джерело тактового сигналу та режим роботи таймера чи лічильника.

Таймер-лічильник може працювати як від внутрішнього тактового генератора через попередній подільник, так і від зовнішнього тактового сигналу, що надходить на вхід T0.

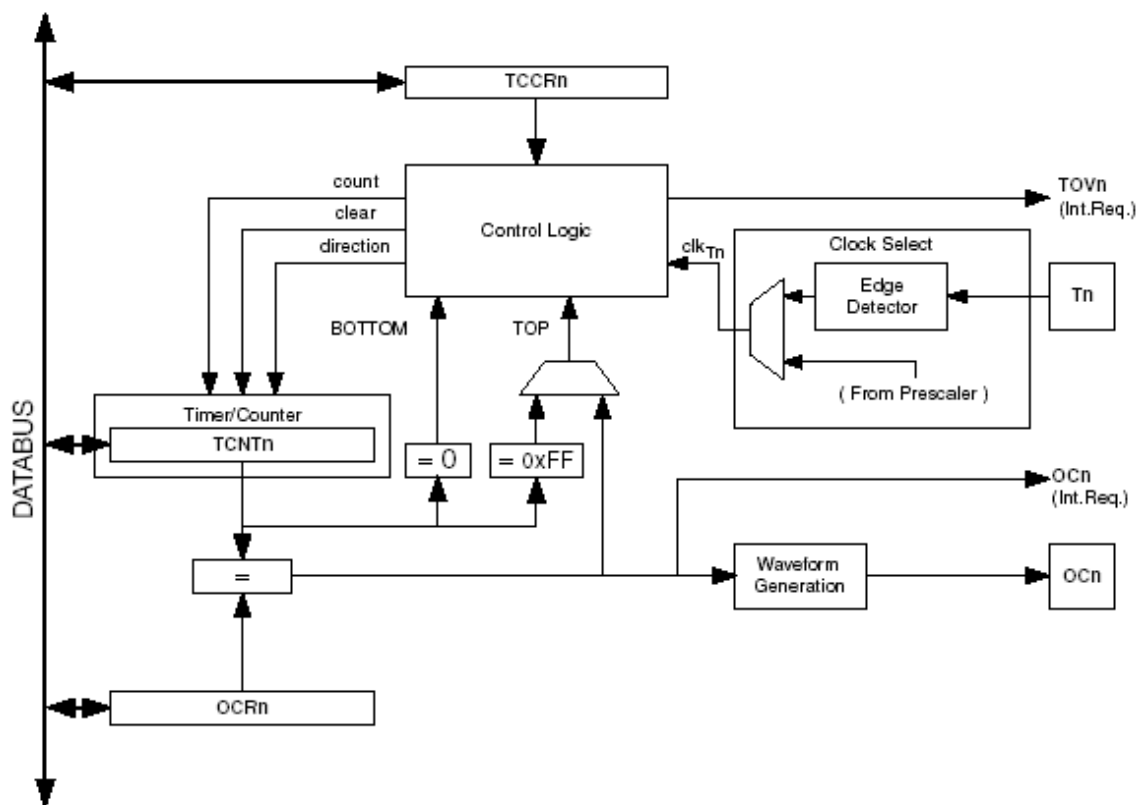


Рис. 121. Спрощена структурна схема таймера-лічильника 0

Схема вибору джерела тактового сигналу передає тактові імпульси вибраного джерела на вхід таймера-лічильника, і кожен імпульс цього сигналу збільшує (або зменшує) значення регістра TCNT0. Якщо не вибрано жодне з джерел тактового сигналу, таймер-лічильник зупиняється. Джерело синхронізації вибирається бітами CS02-0, розташованими у регістрі керування таймера-лічильника TCCR0.

Код, до котрого дорахував таймер-лічильник 0, надходить на цифровий компаратор, де порівнюється із вмістом регістра OCR0. При рівності значень сигнал надходить на запит переривань OC0 та на формувач вихідного сигналу (Waveform Generation). Регістр відліку таймера-лічильника (TCNT0) і регістр порівняння OCR0 є восьмирозрядними.

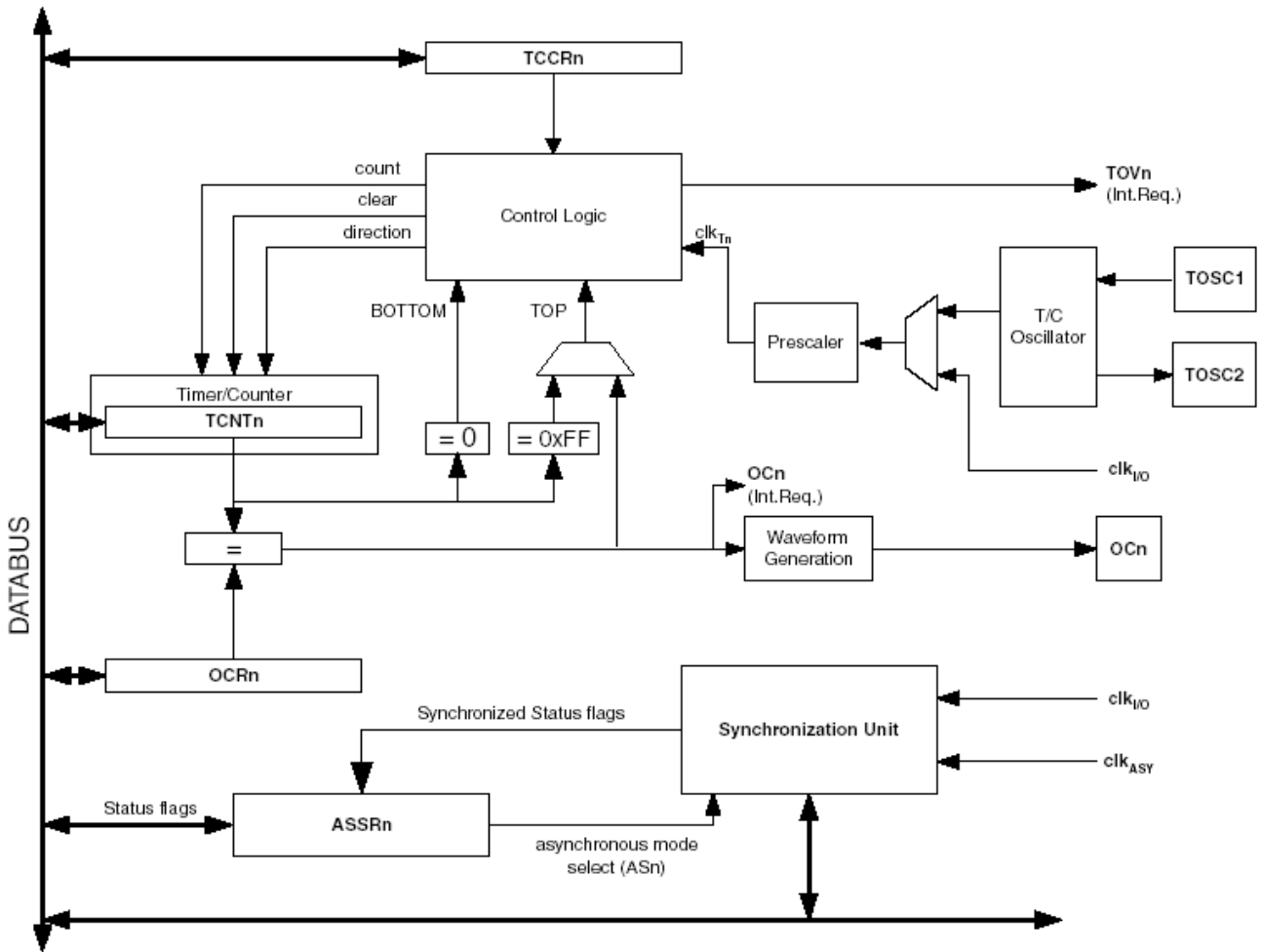


Рис. 122. Спрощена структурна схема таймера-лічильника 2

Кожен із запитів переривань (позначених як Int.Req на рис. 121) індивідуально маскується у регістрі маски таймерів TIMSK і можуть бути перевірені у регістрі запитів переривань таймерів TIFR незалежно від наявності маскування. Регістри TIFR і TIMSK не показані на рис. 121, так як вони є спільними для усіх таймерів.

Структурна схема таймера-лічильника 2 (рис. 122) відрізняється іншою організацією блоку отримання вхідного сигналу і наявністю блоків, що дозволяють асинхронні операції. Будова таймера-лічильника 2 наведена на рис. 122.

У таймер-лічильник 2 введено внутрішній асинхронний генератор (Т/С Oscillator), котрий дозволяє під'єднання як зовнішнього тактового сигналу, так і пряме під'єднання кварцового чи керамічного резонатора. Крім того, з'явився пристрій вибору джерела тактового сигналу, внутрішній попередній подільник

та блок синхронізації (Synchronization Unit) із регістром ASSR2, що містить біти увімкнення асинхронного режиму та біти синхронізації.

Вхід таймера-лічильника 0

У якості вхідного сигналу таймера-лічильника 0 використовується або сигнал із входу T0, або сигнал із попереднього подільника, або тактовий сигнал синхронізації пристроїв вводу-виводу.

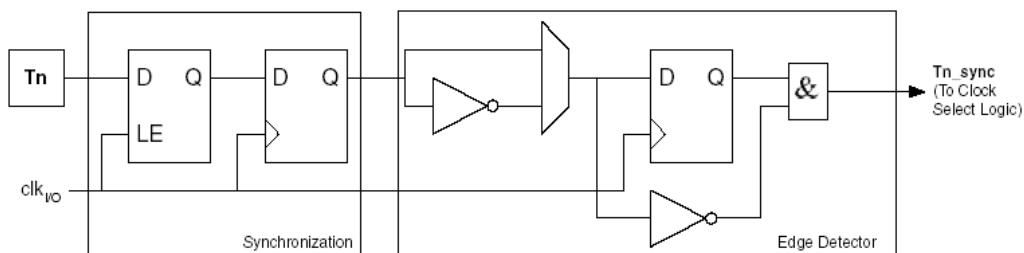


Рис. 123. Вхідне коло лінії T0

Таймер-лічильник може отримувати імпульси із входу T0 і тоді, коли відповідна лінія вводу-виводу конфігурована як вихід. Фронт сигналу T0, за яким здійснюється перемикання таймера-лічильника, вибирається програмним шляхом. Схема формування сигналу на вході лічильника наведена на рис. 123.

Схема складається із двох основних блоків: синхронізатора Synchronizator та детектора фронту Edge Detector. При високому рівні на лінії $clk_{I/O}$ сигнал з лінії пропускається через вхідний тригер і надходить на другий тригер синхронізатора, де фіксується за переднім фронтом наступного імпульсу сигналу $clk_{I/O}$. Така схемотехніка унеможливорює появу на виході синхронізатора імпульсів малої тривалості. Сигнал із синхронізатора надходить на детектор фронту, який формує імпульси тривалістю один період сигналу $clk_{I/O}$ за фронтом сигналу з виходу синхронізатора. Вибір фронту, за яким здійснюється формування імпульсу, здійснюється залежно від стану бітів CS2-0.

Синхронізація та логіка детектора фронту створюють затримку від 2,5 до 3,5 тактових циклів від фронту імпульсу до моменту перемикання лічильника. Також із схеми випливає, що перемикання вхідного тактового сигналу має бути виконане тоді, коли лінія T0 буде стабільною протягом, принаймні, одного

періоду сигналу $f_{clk/I/O}$. З іншого боку, кожна половина періоду зовнішнього сигналу повинна бути більшою, ніж один такт системного тактового сигналу, для забезпечення правильного відбору фронтів. Тому для надійного введення тактового сигналу рекомендується, щоб максимальна частота зовнішнього джерела синхронізації не перевищувала $f_{clk/I/O}/2,5$.

Сигнал із наведеного вище вхідного кола надходить на схему вибору джерела синхронізації, яка наведена на рис. 124.

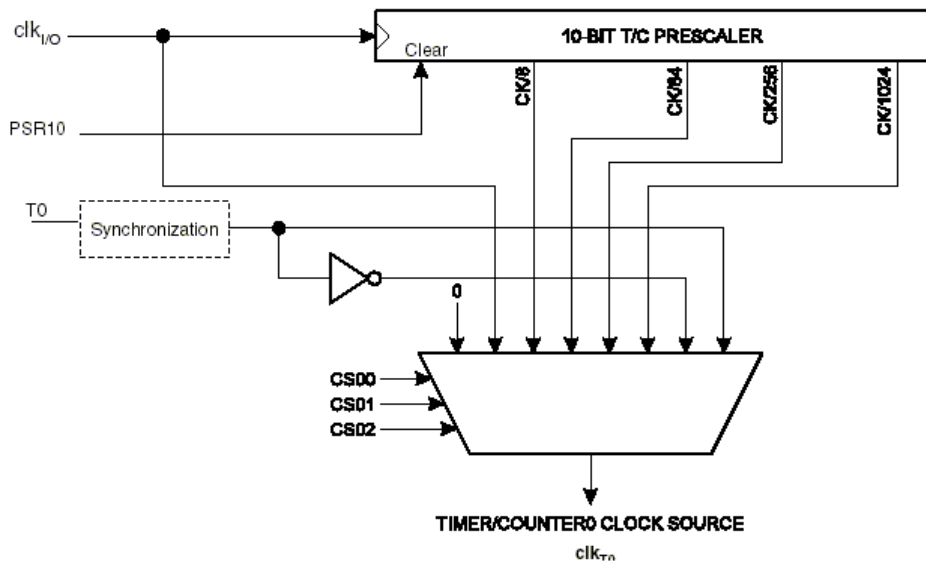


Рис. 124. Схема вибору джерела синхронізації лічильника 0

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 125. Формат регістра TCCR0

Вибір джерела синхронізації здійснюється бітами CS2-0 у регістрі TCCR0 (рис. 125). Біти мають наступне призначення (таблиця 64).

Джерела тактового сигналу для таймера-лічильника 0

CS02	CS01	CS00	Джерело тактового сигналу
0	0	0	Немає сигналу. Лічильник-таймер зупинений
0	0	1	$clk_{I/O}$ (Без попереднього подільника)
0	1	0	Частота з попереднього подільника $clk_{I/O}/8$
0	1	1	Частота з попереднього подільника $clk_{I/O}/64$
1	0	0	Частота з попереднього подільника $clk_{I/O}/256$
1	0	1	Частота з попереднього подільника $clk_{I/O}/1024$
1	1	0	Зовнішнє джерело (лінія T0). Перемикання лічильника за спадом на лінії
1	1	1	Зовнішнє джерело (лінія T0). Перемикання лічильника за переднім фронтом на лінії

Вхід таймера-лічильника 2

Будова входу таймера-лічильника 2 (рис. 126) відрізняється від входу таймера-лічильника 0. Вхід таймера-лічильника 2 дозволяє пряме під'єднання резонатора між виводами TOSC1 та TOSC2. За необхідності замість резонатора на вхід TOSC1 може бути поданий зовнішній тактовий сигнал.

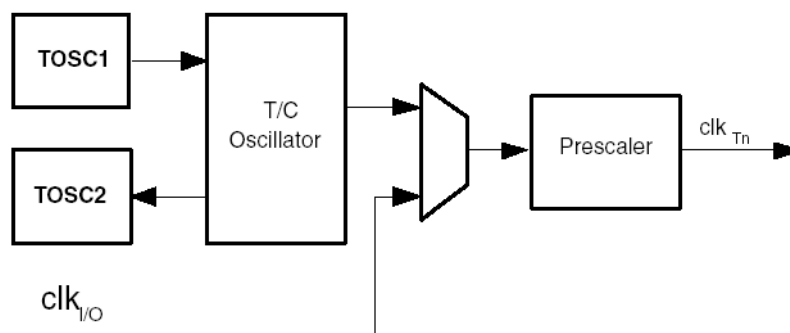


Рис. 126. Вхідне коло таймера-лічильника 2

У будь-якому випадку частота тактового сигналу на лінії має бути меншою у 4 рази, ніж частота основного тактового сигналу.

Вхідне коло містить блок генератора (T/C Oscillator) та попередній подільник (Prescaler). У випадку необхідності, мультиплексор може переключити замість виходу генератора системний тактовий сигнал CLKIO. Переключення джерела тактового сигналу здійснюється бітом AS2 регістра ASSR (рис. 127).

При одиничному значенні біту AS2 джерелом тактового сигналу таймера-лічильника 2 є тактовий генератор, а при нульовому – тактовий сигнал CLKIO. Коли біт AS2 встановлено, виводи PC6 та PC7, що відповідають входам TOSC1 та TOSC2, від'єднуються від порту C та використовуються виключно генератором.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB	ASSR
Read/Write	R	R	R	R	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 127. Регістр ASSR

Сигнал із наведеного вище вхідного кола надходить на схему вибору джерела синхронізації, яка наведена на рис. 128.

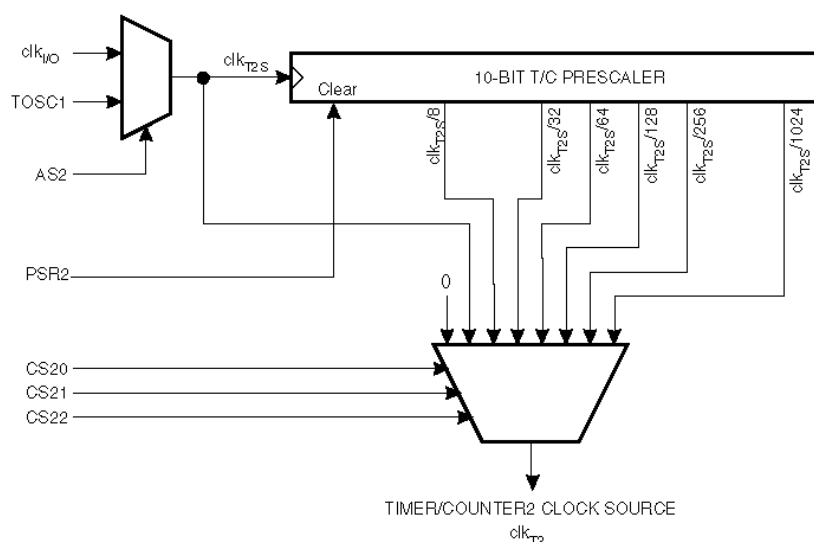


Рис. 128. Вибір джерела тактової частоти таймера-лічильника 2

Bit	7	6	5	4	3	2	1	0	
	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	TCCR2
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 129. Формат регістра TCCR2

Таблиця 65

Джерела тактового сигналу для таймера-лічильника 2

CS02	CS01	CS00	Джерело тактового сигналу
0	0	0	Немає сигналу. Таймер-лічильник зупинений
0	0	1	Тактовий сигнал лічильника Clk без попереднього подільника
0	1	0	Частота з попереднього подільника Clk/8
0	1	1	Частота з попереднього подільника Clk/32
1	0	0	Частота з попереднього подільника Clk/64
1	0	1	Частота з попереднього подільника Clk/128
1	1	0	Частота з попереднього подільника Clk/256
1	1	1	Частота з попереднього подільника Clk/1024

Вибір джерела синхронізації здійснюється бітами CS2-0 у регістрі TCCR2 (рис. 129). Значення бітів регістра відрізняється від бітів регістра TCCR0 і наведені у таблиці 65.

6.10.2.1. Режими роботи таймерів 0 та 2

Таймери-лічильники 0 та 2 можуть використовуватися у таких режимах роботи:

- ◇ нормальному (Normal);
- ◇ зі скиданням за рівністю (CTC);
- ◇ швидкої широтно-імпульсної модуляції (Fast PWM);
- ◇ широтно-імпульсної модуляції із коректною фазою Phase Correct PWM.

Встановлення режиму роботи таймера-лічильника 0 здійснюється за допомогою бітів WGM01:0 у регістрі TCCR0, а таймера-лічильника 2 – за допомогою бітів WGM21:0. Режими роботи ідентичні й наведені у наступній таблиці.

Таблиця 66

Режими роботи таймерів-лічильників 0 та 2

WGMn1	WGMn0	Режим роботи
0	0	Нормальний режим (Normal)
0	1	Режим широтно-імпульсної модуляції із коректною фазою Phase Correct. PWM
1	0	Режим зі скиданням за рівністю (CTC)
1	1	Режим швидкої широтно-імпульсної модуляції (Fast PWM)

У таблиці n дорівнює 0 для таймера-лічильника 0 та дорівнює 2 для таймера-лічильника 2.

Нормальний режим

Нормальний режим (Normal) – найпростіший режим роботи. У цьому режимі таймер-лічильник n (0 чи 2) рахує від заданого значення до значення OFF, потім скидається в нуль і продовжує відлік далі. Для таймера-лічильника 0 при переході зі стану OFF у стан 00 встановлюється в 1 біт TOV0, а для таймера 2 – біт TOV2. У поєднанні з перериванням за переповненням таймера, виклик якого автоматично очищає біт TOV n . Цей режим може використовуватися для підрахунку часових інтервалів чи кількості подій. У даному режимі роботи блок цифрового компаратора може використовуватися довільним чином.

Режим зі скиданням за рівністю

У режимі зі скиданням за рівністю (CTC - Clear Timer on Compare Match) регістр компаратора таймера-лічильника n (n дорівнює 0 чи 2) OCR n використовується для збереження максимального значення, до якого може рахувати лічильник. У режимі CTC лічильник скидається в нуль, коли значення лічильника TCNT n стає рівним OCR n , тобто регістр OCR n визначає верхнє значення для таймера-лічильника n . У цьому режимі роботи ознака переповнення таймера-лічильника TOV n не встановлюється до тих пір, поки таймер-лічильник не перейде зі стану OFF у стан 0. Це може відбутися, якщо значення у лічильнику перевищує значення регістра OCR n .

У даному режимі роботи блок цифрового компаратора може використовуватися для формування сигналу із заданою частотою, що надходить на вихід мікросхеми.

Часова діаграма роботи лічильника у цьому режимі роботи зображена на рис. 130.

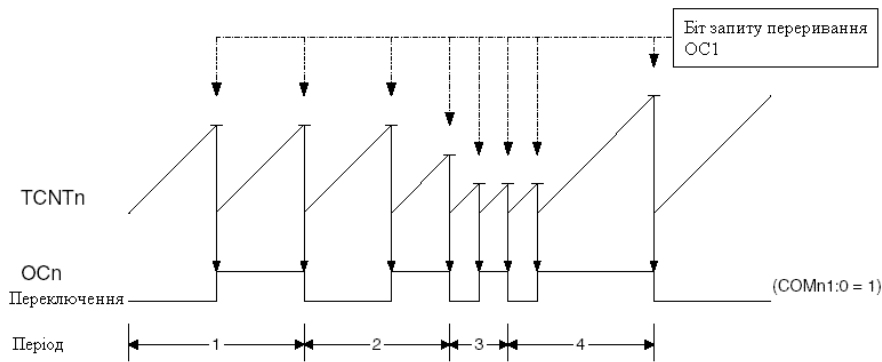


Рис. 130. Режим зі скиданням за рівністю

Для генерації сигналу на виводі в режимі СТС, вихід OCn може бути встановлений у режим перемикачання при кожній рівності (COM n 1 : 0 = 01). Значення на виводі OCn не буде видаватися на вивід порту, поки основна функція порту не налаштована на ввід даних. Максимальна частота сигналу на виході OCn дорівнює $f_{OCn} = f_{CLK}/2$ і досягається, якщо регістр OCRn встановлюється в нуль (0x00). В інших випадках частота сигналу на виході визначається за формулою:

$$f_{OCn} = \frac{f_{clk}}{2 \cdot N \cdot (1 + OCRn)},$$

де N є значенням коефіцієнта поділу попереднього подільника, f_{CLK} – тактова частота лічильника на вході попереднього подільника.

Режим швидкої широтно-імпульсної модуляції

У режимі швидкої широтно-імпульсної модуляції (Fast PWM) лічильник n рахує від 0 до максимального значення. Якщо значення у лічильнику перевищує задане у регістрі OCRn, здійснюється перемикачання відповідного виходу. Вихід повертається у попередній стан при скиданні лічильника. При кожному скиданні лічильника відбувається встановлення ознаки TOVn, а при досягненні значення OCRn – ознаки OCn. У даному режимі роботи центр імпульсу на виході "плаває" відносно початку відліку.

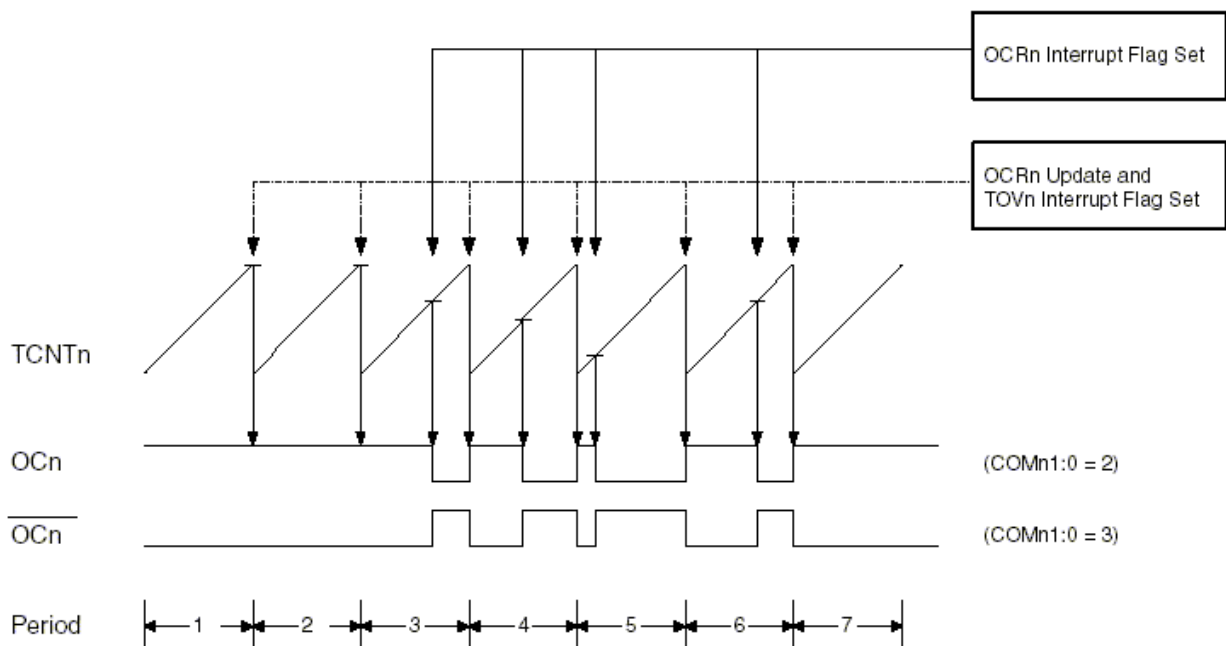


Рис. 131. Режим швидкої широтно-імпульсної модуляції

Режим швидкої широтно-імпульсної модуляції забезпечує високу частоту повторення ШІМ сигналу. Цей режим відрізняється роботою лише за одним схилом. Через роботу за одним схилом робоча частота швидкого режиму ШІМ може бути вдвічі вищою, ніж при роботі у режимі з коректною фазою, що зменшує габарити потрібного фільтра.

Часова діаграма для режиму швидкої ШІМ показана на рис. 131. Значення регістра TCNT0 для ілюстрації показані як гістограма, яка включає режим неінвертованого й інвертованого ШІМ.

Значення частоти вихідного сигналу визначається за допомогою формули

$$f_{OCnPWM} = \frac{f_{clk}}{N \cdot 256},$$

де N є значенням коефіцієнта поділу попереднього подільника, а f_{clk} – тактова частота на вході попереднього подільника.

Режим широтно-імпульсної модуляції з коректною фазою

У режимі широтно-імпульсної модуляції із коректною фазою (PWM, Phase Correct) лічильник n рахує від 0 до максимального значення, а потім – від максимального значення 0FF до 0. Значення 0FF та 0 зберігаються у регістрі

TCNTn протягом одного циклу. Якщо значення у лічильнику перевищує задане у регістрі OCRn, то здійснюється перемикання відповідного виходу. Коли в процесі зворотного відліку значення у лічильнику стане меншим, ніж значення, задане у OCRn, вихід повернеться у початковий стан. У даному режимі роботи положення центру імпульсу на виході відносно початку відліку не залежить від вмісту регістра OCRn. Частота зміни сигналу на виході у цьому режимі роботи менша, ніж у режимі швидкої широтно-імпульсної модуляції, проте, завдяки симетрії імпульсу та постійній частоті, цей метод має свої переваги.

Часова діаграма для режиму широтно-імпульсної модуляції із коректною фазою зображена на рис. 132. Значення TCNTn зображене у вигляді графіка. На діаграмі також наведені неінвертований та інвертований виходи ШІМ.

З рис. 132 випливає, що TOVn встановлюється у момент встановлення мінімального значення у регістрі TCNTn, а біт OCn – у момент рівності значень TCNTn та OCRn. Значення у регістрі OCR0 змінюється у момент набуття регістром TCNTn максимального значення.

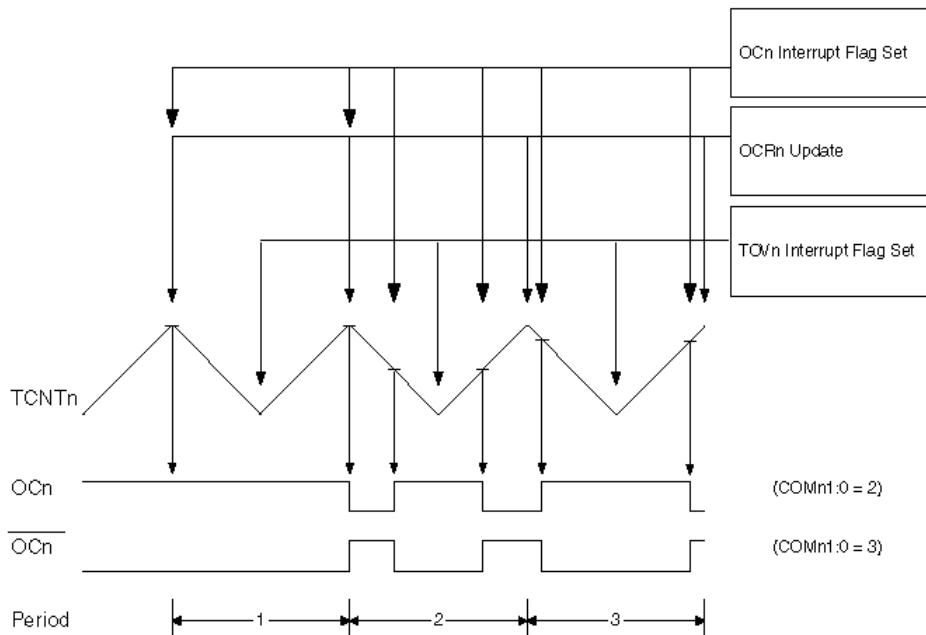


Рис. 132. Часова діаграма режиму широтно-імпульсної модуляції з коректною фазою

Частота вихідного сигналу визначається за формулою

$$f_{OCRnPCPWM} = \frac{f_{clk_I/O}}{N \cdot 510},$$

де N – значення коефіцієнта поділу попереднього подільника.

Схема порівняння

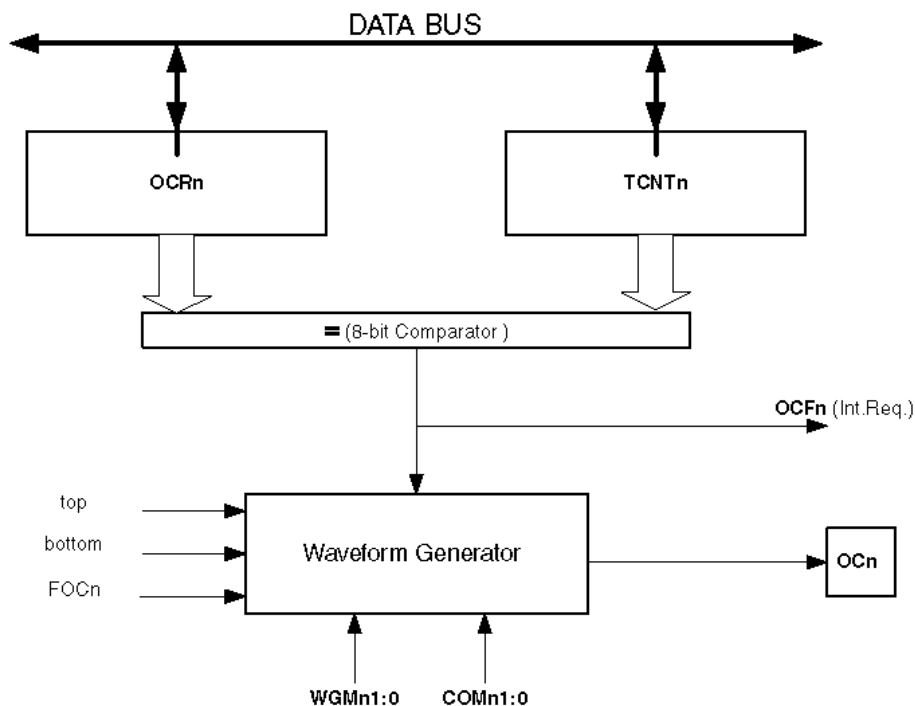


Рис. 133. Схема порівняння 8-бітного таймера-лічильника n (n = 0 або 2)

Схема порівняння лічильників-таймерів 0 та 2 побудована на основі 8-разрядного компаратора, котрий порівнює вміст лічильника TCNTn (n = 0 або 2) з регістром порівняння (OCRn). Кожен раз, коли вміст регістра TCNTn стає рівним OCRn, компаратор сигналізує сигналом рівності. Сигнал рівності перемикає ознаку OCF0, що запускає запит переривання від таймера. Будова схеми порівняння зображена на рис. 133.

При використанні будь-якого режиму ШІМ запис у регістр OCRn буферизований. Для нормального режиму і режиму зі скиданням за рівністю буферизація блокована. Подвійна буферизація синхронізує модифікацію OCRn із досягненням лічильником мінімального чи максимального значення. Синхронізація запобігає утворенню несиметричних імпульсів на виході ШІМ. Усі операції запису у TCNTn блокують будь-яке порівняння на один цикл,

навіть якщо таймер зупинений.

У режимах, не пов'язаних із ШІМ, вихід компаратора може примусово переводитися у стан 1 за допомогою біту FOCn у регістрі TCCRn. Встановлення біту не встановлює ознаку OCFn та не скидає таймер, але стан виводу OC0 буде оновлено як при рівності значень.

Вихід 8-бітного таймера

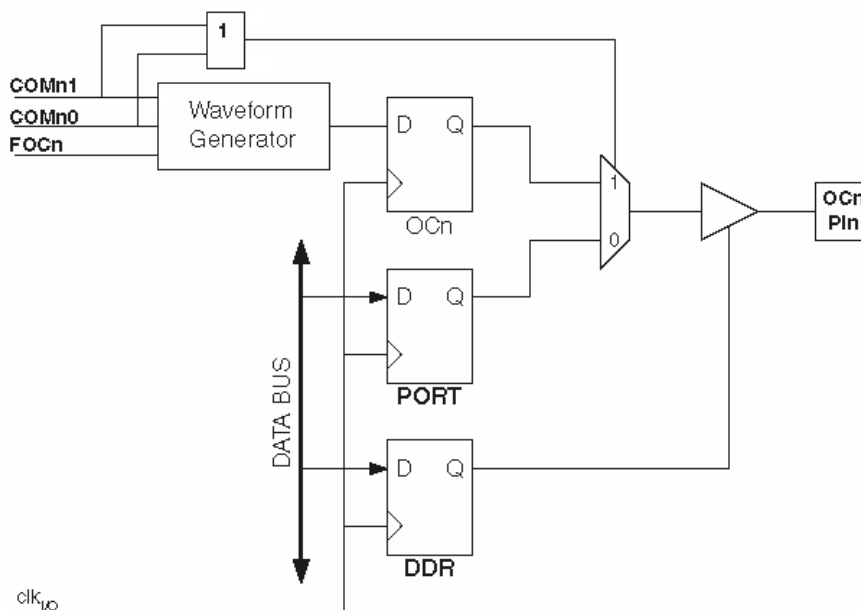


Рис. 134. Спрощена схема виходу таймера 0

Спрощена схемотехніка виходу таймерів-лічильників 0 та 2 однакова і наведена на рис. 134. Якщо будь-який із бітів COMn1:0 ($n = 0$ або 2) встановлений, то основну функцію порту вводу-виводу перекриває біт порівняння OCn.

Таблиця 67

Режими роботи лінії OC0 залежно від стану бітів регістра TCCRn у режимах, не пов'язаних із ШІМ

COMn1	COMn0	Опис роботи лінії OCn
0	0	Нормальний режим роботи порту, біт OCn від'єднаний від порту
0	1	Лінія OCn перемикається при кожному співпаданні
1	0	Лінія OCn скидається при кожному співпаданні
1	1	Лінія OCn встановлюється при кожному співпаданні

Таблиця 68

Режими роботи лінії OC0 залежно від стану бітів регістра TCCRn
у режимі швидкої ШІМ

COMn1	COMn0	Опис роботи лінії OCn
0	0	Нормальний режим роботи порту, біт OCn від'єднаний від порту
0	1	Зарезервовано
1	0	Біт OCn скидається при рівності вмісту таймера (TCNTn) та регістра OCRn, встановлюється при переході лічильника TCNTn до нульового значення (неінвертуючий режим)
1	1	Біт OCn встановлюється при рівності вмісту таймера (TCNTn) та регістра OCRn, скидається при переході лічильника TCNTn до нульового значення (інвертуючий режим)

Таблиця 69

Режими роботи лінії OC0 залежно від стану бітів регістра TCCRn
у режимі з коректною фазою ШІМ

COMn1	COMn0	Опис роботи лінії OCn
0	0	Нормальний режим роботи порту, біт OCn від'єднаний від порту
0	1	Зарезервовано
1	0	Біт OCn скидається при рівності вмісту таймера (TCNTn) та регістра OCRn при зростанні лічильника, встановлюється при рівності вмісту таймера (TCNTn) та регістра OCRn при зростанні лічильника (неінвертуючий режим)
1	1	Біт OCn встановлюється при рівності вмісту таймера (TCNTn) та регістра OCRn при зростанні лічильника, скидається при рівності вмісту таймера (TCNTn) та регістра OCRn при зростанні лічильника (неінвертуючий режим)

Проте регістр напрямку DDR все одно керує напрямком передавання даних лінії порту. Тому для виводу стану порівняння біт у регістрі DDR, що відповідає OCn, має бути переведений у режим виводу незалежно від того, котрий режим роботи таймера- лічильника використовується.

Режим роботи виходу залежить від стану бітів COMn1 COMn0 регістра TCCRn (рис. 125) й обраного режиму роботи. Для всіх режимів роботи режими перемикавання виводу наведені у наступних трьох таблицях.

Асинхронний режим роботи таймера-лічильника 2

Таймер-лічильник 2 може працювати асинхронно, тобто із власним сигналом синхронізації, і не залежити від наявності основного тактового сигналу.

Генератор оптимізовано для використання з годинниковим кварцом на 32 768 Гц. Подача зовнішнього тактового сигналу на лінію TOSC1 може призвести до неправильної роботи таймера-лічильника 1. Частота основного тактового сигналу повинна бути більшою, ніж в чотири рази за частоту генератора.

При роботі в асинхронному режимі роботи під час запису в один з регістрів TCNT2, OCR2 чи TCCR2 значення передається у тимчасовий регістр, окремий для кожного регістра, і фіксується через два передніх фронти сигналу з лінії TOSC1. А тому запис TCNT2 не заважає запису, наприклад, в OCR2. З іншого боку, запис нового значення не має перекривати процес запису попереднього значення. Готовність регістрів до прийому значень може бути перевірена за допомогою регістра ASSR.

При використанні енергозберігаючих режимів і використанні можливості "пробудження" від переривання від таймера 2, що працює асинхронно, слід віддавати команду переходу у енергозберігаючий режим лише за наявності двох умов:

1. Таймер-лічильник 2 не має встановлених бітів зайнятості у регістрі ASSR.

2. З моменту надходження попереднього запиту переривання від таймера-лічильника 2 надійшов хоча б один тактовий імпульс із лінії TOSC1.

Якщо програма "не знає" необхідного часу до надходження наступного імпульсу на лінії TOSC1, можна скористуватися таким алгоритмом:

1. Записати значення у регістр TCCR2, TCNT2 або OCR2.
2. Зачекати, доки скинеться відповідний біт очікування у регістрі ASSR.
3. Перейти у визначений режим зниженого енергоспоживання.

При роботі в асинхронному режимі генератор таймера-лічильника 2 працює завжди, за винятком режимів Power-down та Standby, після ввімкнення живлення або виходу із режимів від'єднання живлення і режиму очікування. Після ввімкнення живлення або виходу із режимів Power-down та Standby необхідно до 1 секунди для стабілізації частоти опорного тактового сигналу. Вміст всіх регістрів таймера-лічильника 2 при цьому можна вважати втраченим.

Читання регістра TCNT2 відразу після пробудження з енергозберігаючих режимів може дати неправильний результат. Щоб уникнути цього, необхідно дочекатися надходження ще одного переднього фронту на лінії TOSC1 або записати у регістр OCR2 чи TCCR2 та дочекатися скидання відповідної ознаки зайнятості.

У процесі перемикання між асинхронним і синхронним режимами роботи вміст регістрів TCNT2, OCR2 і TCCR2 може бути пошкоджений. Безпечна процедура для перемикання джерела тактового сигналу є такою:

1. Вимкнути переривання від таймера, скинувши біти OCIE2 і TOIE2.
2. Обрати джерело синхронізації, задавши значення AS2.
3. Записати нові значення у TCNT2, OCR2 і TCCR2.
4. В асинхронному режимі роботи дочекатися скидання бітів зайнятості у регістрі ASSR.
5. Скинути ознаки переривань від таймера-лічильника 2.
6. Дозволити переривання від таймера, якщо це необхідно.

6.10.2.2. Регістри, пов'язані з таймером-лічильником 0

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 135. Керуючий регістр таймера-лічильника 0 TCCR0

Керуючий регістр таймера-лічильника 0 TCCR0 призначений для задавання режиму роботи мікросхеми. Біти регістра описані вище.

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 136. Регістр відліку TCNT0

Регістр відліку TCNT0 дає прямий доступ як для читання, так і для запису 8-розрядного значення таймера-лічильника 0. Запис у регістр TCNT0 блокує порівняння на наступний період синхронізації.

Bit	7	6	5	4	3	2	1	0	
	OCR0[7:0]								OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 137. Регістр вихідного компаратора OCR0

Регістр вихідного компаратора OCR0 містить 8-розрядне значення, з котрим порівнюється вихідне значення таймера, та можливе формувати значення на лінії OC0.

6.10.2.3. Регістри, пов'язані з таймером-лічильником 2

Bit	7	6	5	4	3	2	1	0	
	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	TCCR2
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 138. Керуючий регістр таймера-лічильника 2 TCCR2

Керуючий регістр таймера-лічильника 2 TCCR2 призначений для задавання режиму роботи мікросхеми. Біти регістра описані вище. При читанні TCCR2 читається значення в регістрі тимчасового зберігання.

Bit	7	6	5	4	3	2	1	0	
	TCNT2[7:0]								TCNT2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 139. Регістр відліку TCNT2

Регістр відліку TCNT2 дає прямий доступ як для читання, так і для запису 8-розрядного значення таймера-лічильника 2. Запис у регістр TCNT2 блокує порівняння на наступний період синхронізації. При читанні TCNT2 читається фактичне значення таймера не залежно від режиму роботи.

Bit	7	6	5	4	3	2	1	0	
	OCR2[7:0]								OCR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 140. Регістр вихідного компаратора OCR2

Регістр вихідного компаратора OCR2 містить 8-разрядне значення, з котрим порівнюється вихідне значення таймера та стає можливим формувати значення на лінії OC2. При читанні OCR2 читається значення в регістрі тимчасового зберігання.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB	ASSR
Read/Write	R	R	R	R	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 141. Регістр асинхронних операцій

Регістр ASSR використовується для задавання та контролю за роботою таймера-лічильника 2 в асинхронному режимі роботи.

Біт AS2 використовується для задавання джерела синхронізації. Нульове значення вказує на синхронізацію від основного тактового сигналу, одиничне – на синхронізацію від лінії TOSC1 та приєднаного до неї генератора. При зміні значення AS2 вміст регістрів TCNT2, OCR2 і TCCR2 може бути пошкоджений.

Біт TCN2UB вказує на зайнятість таймера оновленням регістра TCNT2. Біт автоматично встановлюється в одиницю при запису у регістр TCNT2 при роботі в асинхронному режимі й автоматично скидається в нуль, коли оновлення регістра TCNT2 завершено.

Біт OCR2UB вказує на зайнятість таймера оновленням регістра OCR2. Біт автоматично встановлюється в одиницю при запису у регістр OCR2 при роботі в асинхронному режимі й автоматично скидається в нуль, коли оновлення регістра OCR2 завершено.

Біт TCR2UB вказує на зайнятість таймера оновленням регістра TCCR2. Біт автоматично встановлюється в одиницю при запису у регістр TCCR2 при роботі в асинхронному режимі й автоматично скидається в нуль, коли оновлення регістра TCCR2 завершено.

6.10.2.4. Біти загальних регістрів, використовувані таймерами 0 та 2

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 142. Регістр маски переривань TIMSK

Для налаштування лічильника, крім регістрів самого лічильника, використовуються регістри, що є загальними для всіх таймерів.

Регістр маски переривань TIMSK є загальним для усіх таймерів. Для таймера 0 використовуються біти OCIE0 та TOIE0, а для таймера 2 – біти OCIE2 та TOIE2.

Біт OCIE0, встановлений в 1, дозволяє переривання від блоку порівняння таймера-лічильника 0 (біт OCF0)

Біт TOIE0, встановлений в 1, дозволяє переривання при переповненні таймера-лічильника 0 (біт TOV0).

Біт OCIE2, встановлений в 1, дозволяє переривання від блоку порівняння таймера-лічильника 2 (біт OCF2).

Біт TOIE2, встановлений в 1, дозволяє переривання при переповненні таймера-лічильника 2 (біт TOV2).

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 143. Регістр переривань TIFR

У регістрі переривань таймерів TIFR відображається стан переривань від усіх таймерів мікросхеми. До таймера-лічильника 0 відносяться біти OCF0 та TOV0, а до таймера-лічильника 2 – біти OCF2 та TOV2.

Біти OCF0 та OCF2 відображають стан запиту переривань від блоку порівняння таймера 0 чи 2, а біти TOV0 та TOV2 – переривання при переповненні таймерів 0 чи 2. Біти скидаються автоматично при виклику відповідного переривання, крім того, вони можуть скидатися і програмним шляхом.

6.10.3. 16-розрядний таймер-лічильник 1

Таймер-лічильник T/C1 – це універсальний шістнадцятибітний лічильник з двома модулями порівняння і одним модулем захоплення, а також підтримкою функції широтно-імпульсної модуляції ШІМ (PWM) зі змінною частотою модуляції. Він дозволяє формувати задані проміжки часу для роботи в режимі реального часу, а також може служити генератором сигналів. Вхідним сигналом пристрою може бути як зовнішній сигнал з виводу мікросхеми, так і сигнал із тактового генератора, котрий може бути пропущений через попередній подільник.

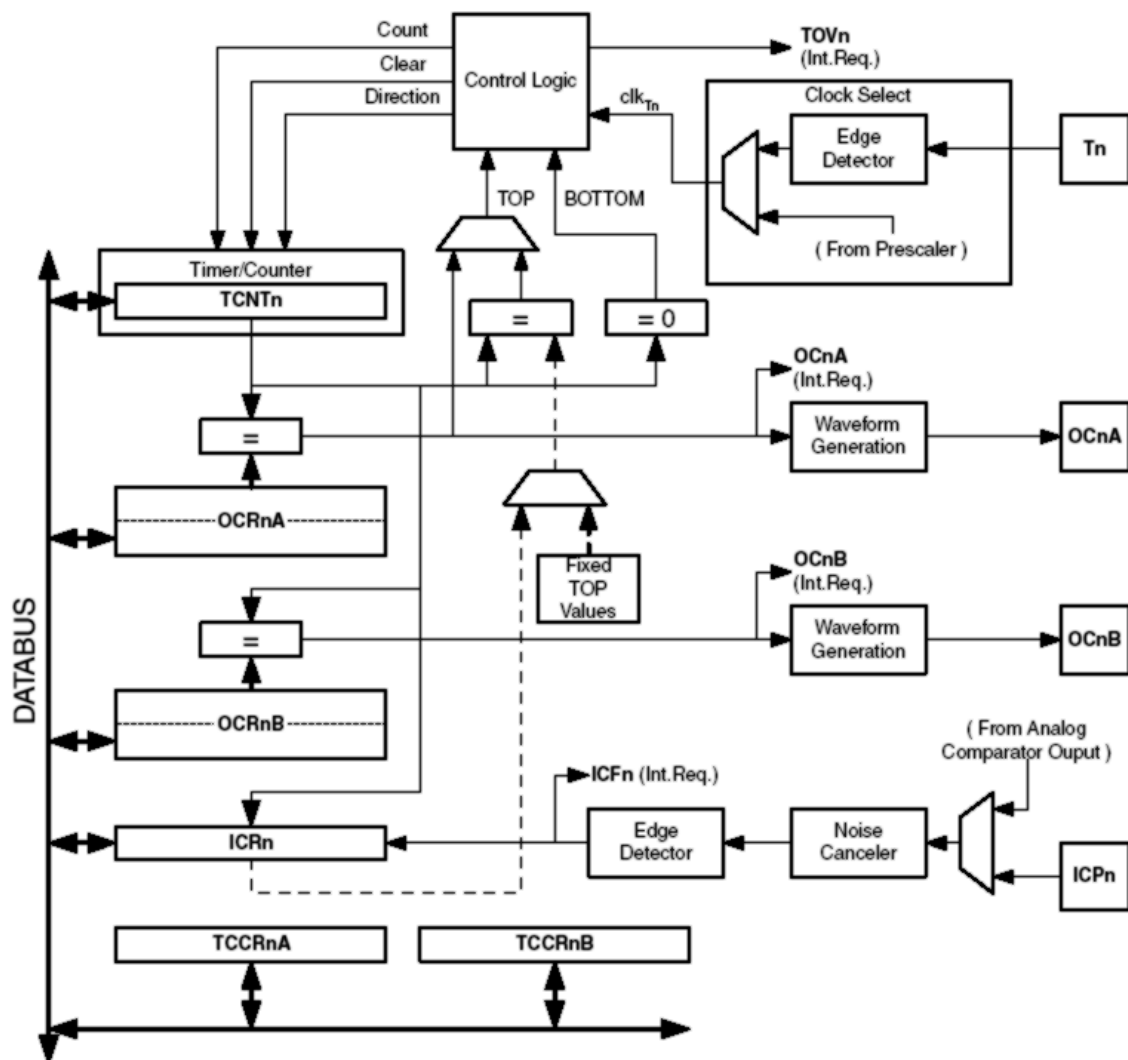


Рис. 144. Спрощена схема таймера-лічильника 1

Основні особливості таймера:

- ◇ наявність двох модулів порівняння, що можуть формувати два сигнали на відповідні виходи та два запити переривань від них;
- ◇ можливість подвійної буферизації при запису в регістри порівняння;
- ◇ скидання таймера при рівності значення заданому;
- ◇ наявність симетричного широтно-імпульсного модулятора з коректною фазою та частотою;
- ◇ змінний період у режимі ШІМ;
- ◇ чотири незалежні джерела переривання (TOV1, OC1A, OC1B, ICF0);
- ◇ наявність модуля захоплення.

Імпульси, які підраховуються, надходять на блок керування від блоку вибору сигналу синхронізації (Clock Select), котрий визначає джерело тактового сигналу та режим роботи таймера чи лічильника. Таймер-лічильник може працювати як від внутрішнього тактового генератора через попередній дільник, так і від зовнішнього тактового сигналу, що надходить на вхід T1. Схема вибору джерела тактового сигналу передає тактові імпульси вибраного джерела на вхід таймера-лічильника і кожен імпульс цього сигналу збільшує (або зменшує) значення регістра TCNT1.

Якщо не вибране жодне з джерел тактового сигналу, таймер-лічильник зупиняється. Джерело синхронізації вибирається бітами CS12-0, розташованими у регістрі керування таймера-лічильника TCCR1.

Код, до котрого дорахував таймер-лічильник 1, надходить на два цифрових компаратори, де порівнюються із вмістом регістрів OCR1A та OCR1B. При рівності значень сигнал надходить на запит переривань OC1A, OC1B та на відповідні формувачі вихідних сигналів (Waveform Generation). Поточне значення таймера-лічильника може бути зафіксоване за зовнішнім сигналом, що здійснюється модулем захоплення (Input Capture).

Регістр відліку таймера-лічильника (TCNT1), регістри порівняння OCR1A

та OCR1A, а також регістр захоплення ICR1 є 16-бітними. Максимальне значення відліку, до якого може рахувати таймер-лічильник 1, може бути фіксованим і рівним 0FF, 1FF, 3FF, 0FFFF, або зберігатись у регістрі OCR1A чи ICR1.

Кожен із запитів переривань (позначених як Int.Req. на рис. 144) індивідуально маскується у регістрі маски таймерів TIMSK, і може бути перевірений у регістрі запитів переривань таймерів TIFR незалежно від наявності маскування. Регістри TIFR і TIMSK не показані на рис. 144, так як вони є спільними для усіх таймерів мікросхеми.

Вхід лічильника-таймера 1

У якості вхідного сигналу таймера-лічильника 1 використовується або сигнал із входу T1, або сигнал із попереднього подільника, або тактовий сигнал синхронізації пристроїв вводу-виводу. Таймер-лічильник може отримувати імпульси із входу T1 і тоді, коли відповідна лінія вводу-виводу конфігурована як вихід.

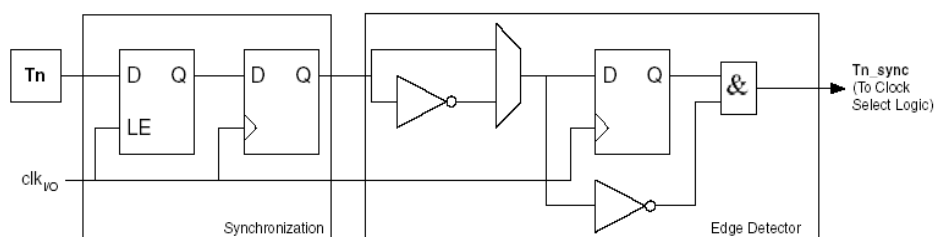


Рис. 145. Вхідне коло лінії T1

Фронт сигналу T1, за яким здійснюється перемикання таймера-лічильника, вибирається програмним шляхом. Схема формування сигналу на вході лічильника наведена на рис. 145.

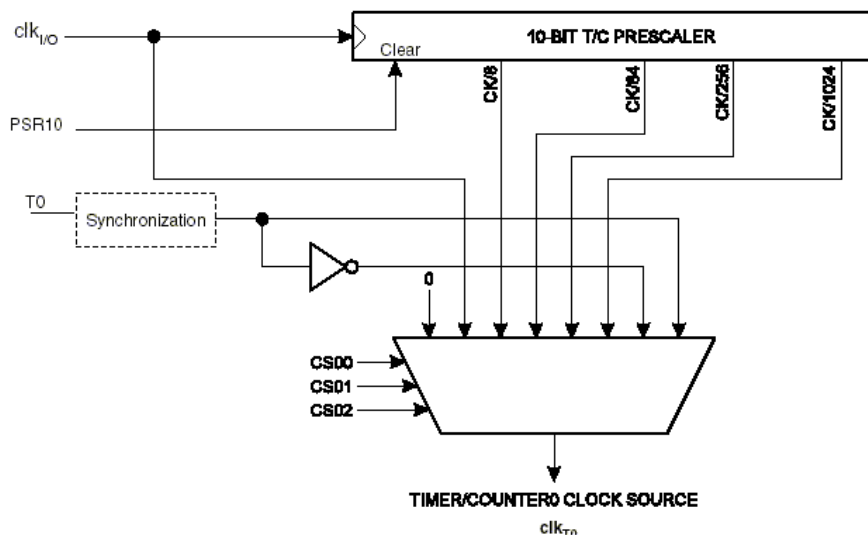


Рис. 146. Схема вибору джерела синхронізації лічильника 1

Схема складається із двох основних блоків: синхронізатора Synchronizator та детектора фронту Edge Detector. При високому рівні на лінії $clk_{I/O}$ сигнал з лінії пропускається через вхідний тригер і надходить на другий тригер синхронізатора, де фіксується за переднім фронтом наступного імпульсу сигналу $clk_{I/O}$. Така схемотехніка унеможливорює появу на виході синхронізатора імпульсів малої тривалості. Сигнал із синхронізатора надходить на детектор фронту, котрий формує імпульси тривалістю один період сигналу $clk_{I/O}$ за фронтом сигналу з виходу синхронізатора. Вибір фронту, за яким здійснюється формування імпульсу, залежить від стану бітів CS12-0.

Синхронізація та логіка детектора фронту створюють затримку від 2,5 до 3,5 тактових циклів від фронту імпульсу до моменту перемикавання лічильника. Зі схеми також випливає, що перемикавання вхідного тактового сигналу має бути виконане тоді, коли лінія T0 буде стабільною протягом, принаймні, одного періоду сигналу $fclk_{I/O}$. З іншого боку, кожна половина періоду зовнішнього сигналу повинна бути більшою, ніж один такт системного тактового сигналу, для забезпечення правильного відбору фронтів. Тому для надійного введення тактового сигналу рекомендується, щоб максимальна частота зовнішнього джерела синхронізації не перевищувала $fclk_{I/O}/2,5$.

Сигнал із наведеного вище вхідного кола надходить на схему вибору джерела синхронізації, схема котрого наведена на рис. 146.

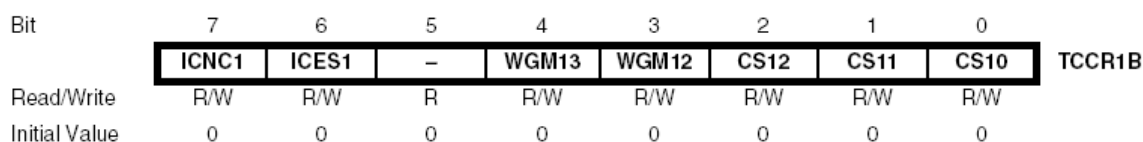


Рис. 147. Формат регістра TCCR1B

Вибір джерела синхронізації здійснюється бітами CS2-0 у регістрі TCCR1B (рис. 147). Біти мають наступне призначення (таблиця 70).

Таблиця 70

Джерела тактового сигналу для таймера-лічильника 1

CS12	CS11	CS10	Джерело тактового сигналу
1	2	3	4
0	0	0	Немає сигналу. Таймер-лічильник зупинений
0	0	1	clk _{I/O} (Без попереднього подільника)
0	1	0	Частота з попереднього подільника clk _{I/O} /8
0	1	1	Частота з попереднього подільника clk _{I/O} /64
1	0	0	Частота з попереднього подільника clk _{I/O} /256
1	0	1	Частота з попереднього подільника clk _{I/O} /1024
1	1	0	Зовнішнє джерело (лінія T1). Перемикання лічильника за спадом на лінії
1	1	1	Зовнішнє джерело (лінія T1). Перемикання лічильника за переднім фронтом на лінії

Блок захоплення

Таймер-лічильник 1 включає блок захоплення, що дозволяє легко визначати час появи зовнішньої події. Поточне значення лічильника може бути зафіксоване у регістрі модуля захоплення. Для фіксації значення використовується стробуючий сигнал, що може надходити від аналогового компаратора (Analog Comparator) чи від лінії ICP1.

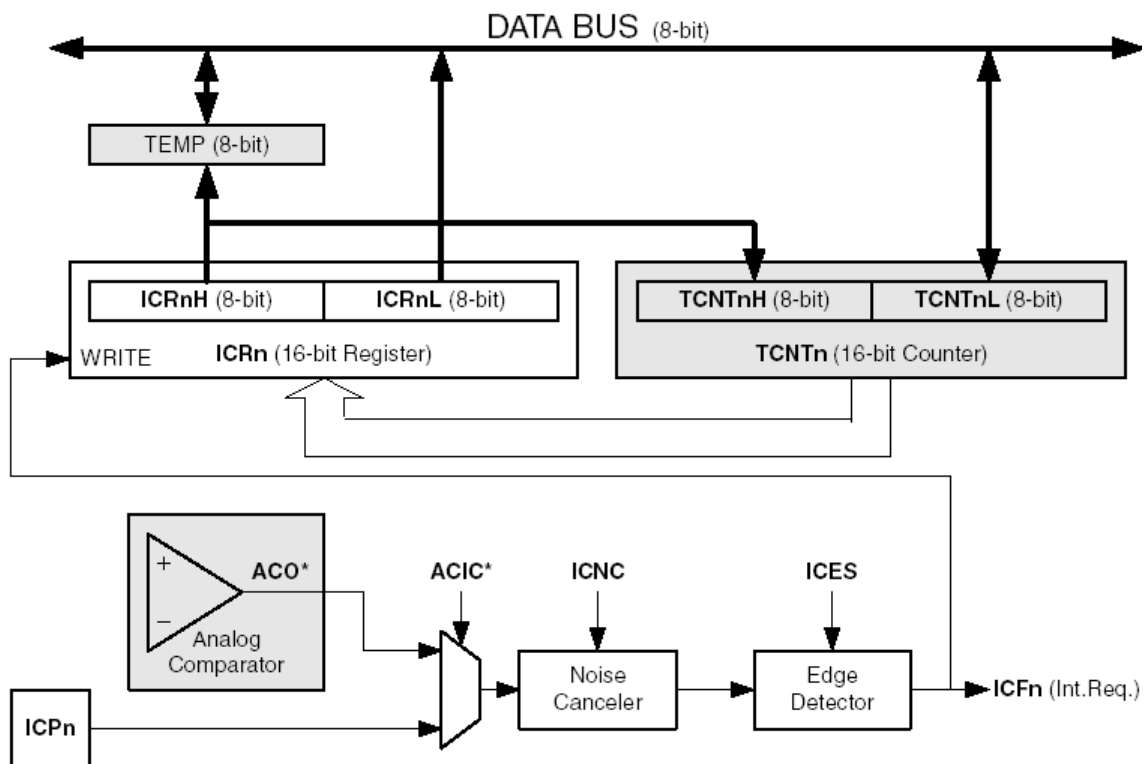


Рис. 148. Структурна схема блоку захоплення

Мітки часу, що генеруються блоком, можуть використовуватися для обчислення періоду та параметрів сигналу, або для створення журналу подій. Будова блоку захоплення наведена на рис. 148. Елементи схеми, які не є частиною блоку, наведені суцільною лінією.

Вибір джерела стробуючого сигналу для керування блоком визначається бітом ACIC у регістрі ACSR. При запису логічної одиниці цей біт вмикає функцію захоплення таймера-лічильника 1 від аналогового компаратора. При записі логічного нуля захоплення здійснюється за станом лінії ICP1. Вихід компаратора чи лінія ICP1 під'єднуються до цифрового фільтра (Noise Canceler) та детектора перепаду (Edge Detector), та керують роботою регістра ICR1, що фіксує поточне значення лічильника TCNT1.

Детектор фронту ідентичний такому ж детектору для входу T1. Цифровий фільтр призначений для усунення багатократних перемикань входу та підвищує перешкодозахищеність за допомогою простої схеми цифрової фільтрації: вхід контролюється протягом чотирьох тактів і, якщо у всіх чотирьох тактах сигнал є однаковим, вихід перемикається у новий стан. Цифровий фільтр вмикається

установкою біту ICNC1 у регістрі керування TCCR1B. Увімкнений цифровий фільтр вводить затримку у чотири додаткові такти перед оновленням ICR1.

Читання регістра ICR1 програмним шляхом можливе у будь-якому режимі роботи. Запис у регістр ICR1 можливе лише при використанні режиму генерації сигналу, який, в свою чергу, використовує регістр ICR1 для визначення максимального значення лічильника.

У момент фіксації значення в регістрі ICR1 створюється запит переривання ICF1. Якщо ж встановлено біт TICIE1 = 1 у регістрі TIMSK, ознака захоплення входу формує переривання вхідного захоплення, а біт ICF1 при цьому автоматично очищається при переході до обслуговування переривання. Крім того, біт ICF1 може бути скинутий програмно за допомогою команд вводу-виводу.

Блоки порівняння

Мікросхема має два блоки порівняння, названі Output Compare Unit A та Output Compare Unit B. Блок порівняння A також дозволяє встановити максимальне значення відліку лічильника у деяких режимах роботи. Будова кожного із блоків порівняння наведена на рис. 149, де n позначає номер лічильника, а x – блок A чи B. Елементи схеми, що не є безпосередньо частиною блоку порівняння, виділені суцільною лінією.

Кожен 16-бітний компаратор постійно порівнює вміст лічильника TCNT1 з виходом регістра компаратора OCR1x (OCR1A чи OCR1B). Якщо значення TCNT та OCR1x співпадають, видається сигнал рівності, що встановлює ознаку OCF1A чи OCF1B у наступному такті таймера. У випадку, коли дозволено відповідне переривання від блоку порівняння OCIE1x=1 (OCIE1A=1 або OCIE1B=1), вихід компаратора генерує запит переривання. Ознака OCF1x автоматично очищається при старті переривання. Крім того, біти OCF1x можуть бути скинуті програмно.

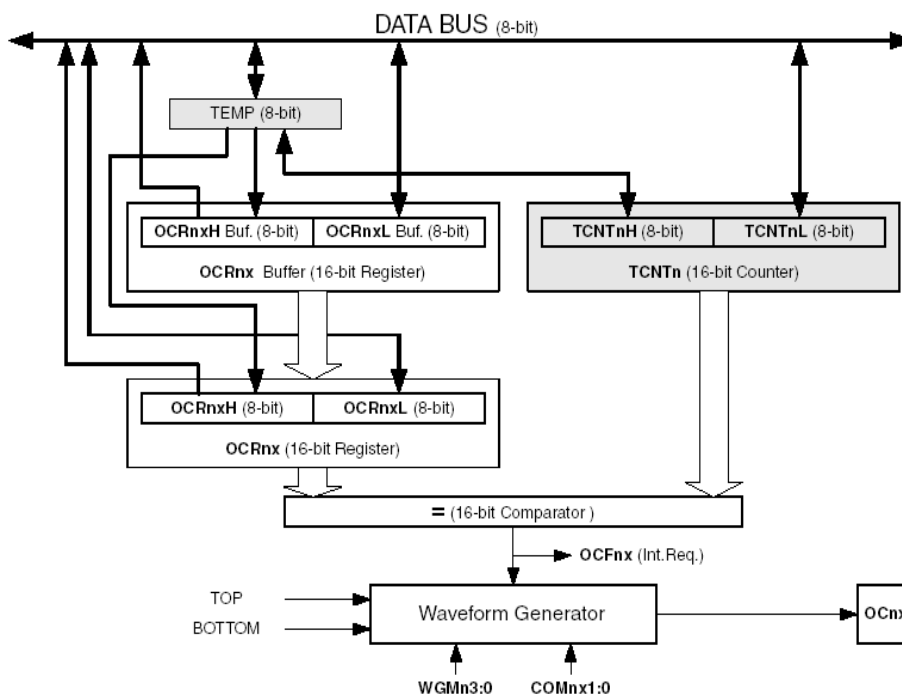


Рис. 149. Блок порівняння таймера-лічильника 1

Сигнал із блоку порівняння надходить на генератор вихідного сигналу (Waveform Generator) і використовується залежно від режиму роботи (бітів WGM13: 0) і бітів режиму виводу COM1x1: 0.

У випадку необхідності у режимах без ШІМ на виході компаратора може бути примусово отриманий сигнал рівності, але при цьому не буде викликане переривання, не буде встановлена ознака рівності та не буде скинутий лічильник. Примусове встановлення біту компаратора здійснюється записом одиниці у біти FOC1A та FOC1B. Біти примусово скидаються у наступному машинному такті, тобто при наступному читанні біти будуть прочитані як 0.

6.10.3.1. Режими роботи таймера 1

Таймер-лічильник 1 може використовуватися у таких режимах роботи:

- ◇ нормальному (Normal);
- ◇ зі скиданням за рівністю (CTC);
- ◇ швидкої широтно-імпульсної модуляції (Fast PWM);
- ◇ широтно-імпульсної модуляції з коректною фазою Phase Correct PWM;
- ◇ широтно-імпульсної модуляції із коректною фазою та частотою Phase and Frequency Correct PWM.

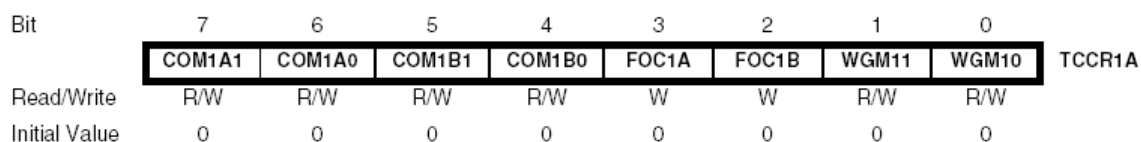


Рис. 150. Формат регістра TCCR1A

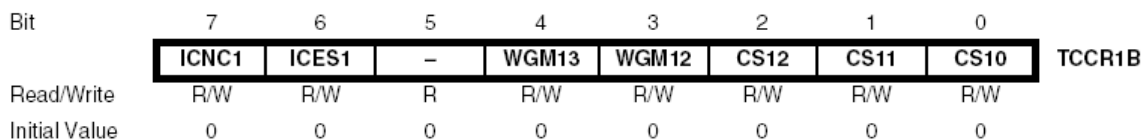


Рис. 151. Формат регістра TCCR1B

Таблиця 71

Режими таймера-лічильника 1

WGM13- WGM10				Режим роботи таймера-лічильника	Максимальне значення	Завантаження нового значення у регістр OCR1x	Умова встановлення біту TOV1
13	12	11	10				
1	2	3	4	5	6	7	8
0	0	0	0	Нормальний	0xFFFF	Відразу при перезавантаженні регістра OCR1x	При досягненні лічильником значення 0xFFFF
0	0	0	1	Широтно-імпульсної модуляції з коректною фазою Phase Correct PWM, 8-біт	0x00FF	При досягненні лічильником значення 0x00FF	При досягненні лічильником значення 0x0000
0	0	1	0	Широтно-імпульсної модуляції із коректною фазою Phase Correct PWM, 9-біт	0x01FF	При досягненні лічильником значення 0x01FF	При досягненні лічильником значення 0x0000
0	0	1	1	Широтно-імпульсної модуляції із коректною фазою Phase Correct PWM., 10-біт	0x03FF	При досягненні лічильником значення 0x03FF	При досягненні лічильником значення 0x0000
0	1	0	0	Зі скиданням за рівністю (CTC)	OCR1A	Відразу при перезавантаженні регістра OCR1x	При досягненні лічильником значення 0xFFFF
0	1	0	1	Швидкої широтно-імпульсної модуляції (Fast PWM), 8-біт	0x00FF	При досягненні лічильником значення 0x0000	При досягненні лічильником значення 0x00FF

Закінчення табл. 71							
0	1	1	0	Швидкої широтно-імпульсної модуляції (Fast PWM), 9-біт	0x01FF	При досягненні лічильником значення 0x0000	При досягненні лічильником значення 0x01FF
0	1	1	1	Швидкої широтно-імпульсної модуляції (Fast PWM), 10-біт	0x03FF	При досягненні лічильником значення 0x0000	При досягненні лічильником значення 0x03FF
1	0	0	0	Широтно-імпульсної модуляції з коректною фазою та частотою Phase and Frequency Correct PWM	Значення із ICR1	При досягненні лічильником значення 0x0000	При досягненні лічильником значення 0x0000
1	0	0	1	Широтно-імпульсної модуляції з коректною фазою та частотою Phase and Frequency Correct PWM	Значення із OCR1A	При досягненні лічильником значення 0x0000	При досягненні лічильником значення 0x0000
1	0	1	0	Широтно-імпульсної модуляції з коректною фазою Phase Correct PWM	Значення із ICR1	При досягненні лічильником значення ICR1	При досягненні лічильником значення 0x0000
1	0	1	1	Широтно-імпульсної модуляції з коректною фазою Phase Correct PWM	Значення із OCR1A	При досягненні лічильником значення OCR1A	При досягненні лічильником значення 0x0000
1	1	0	0	Зі скиданням за рівністю (CTC)	Значення із ICR1	Відразу при перезавантаженні регістра ICR1x	При досягненні лічильником значення 0x0FFF
1	1	0	1	Зарезервовано	–	–	
1	1	1	0	Швидкої широтно-імпульсної модуляції (Fast PWM)	Значення із ICR1	При досягненні лічильником значення 0x0000	При досягненні лічильником значення ICR1
1	1	1	1	Швидкої широтно-імпульсної модуляції (Fast PWM)	Значення із OCR1A	При досягненні лічильником значення 0x0000	При досягненні лічильником значення OCR1A

Встановлення режиму роботи таймера-лічильника 1 здійснюється за допомогою бітів WGM13:0 у регістрах TCCR1A (рис. 150) та TCCR1B (рис. 151), як показано у наступній таблиці.

Нормальний режим

Нормальний режим (Normal) – найпростіший. У ньому лічильник рахує від заданого значення до значення 0FFFF, потім скидається в нуль і продовжує відлік далі. При переході зі стану 0FFFF у стан 0000 встановлюється в 1 біт TOV0. У поєднанні з перериванням за переповненням таймера, виклик якого автоматично очищає біт TOV0, цей режим може використовуватися для підрахунку часових інтервалів чи кількості подій тощо. У даному режимі роботи блок цифрового компаратора може використовуватися довільним чином.

Режим зі скиданням за рівністю

У режимі зі скиданням за рівністю (CTC – Clear Timer on Compare Match) регістр компаратора OCR1A або регістр захоплення ICR1 використовують для збереження максимального значення, до якого може рахувати лічильник. У режимі CTC лічильник скидається в нуль, коли значення лічильника TCNT0 стає рівним OCR1A чи ICR1. У цьому режимі роботи ознака переповнення таймера-лічильника TOV0 не встановлюється до тих пір, поки таймер-лічильник не перейде зі стану 0FFFF у стан 0, що може відбутися, якщо значення у лічильнику перевищує значення обраного регістра.

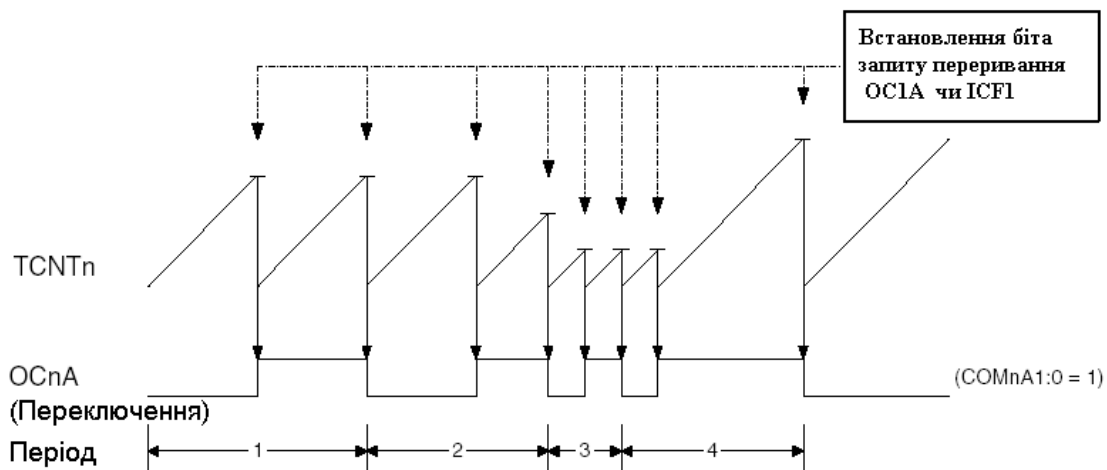


Рис. 152. Режим зі скиданням за рівністю

У даному режимі роботи блок цифрового компаратора може використовуватися для формування сигналу із заданою частотою, що

надходить на вихід мікросхеми. Часова діаграма роботи лічильника у цьому режимі роботи зображена на рис. 152.

Для генерації сигналу на виході у CTC-режимі вихід OC1A може бути встановлений для перемикавання логічного рівня на кожному порівнянні при встановленні бітів режим виводу (COM1A1: 0 = 1). Проте значення OC1A не буде видно на виводі мікросхеми, поки вивід не налаштований на вхід (DDR_OC1A = 1). Максимальна частота сигналу на виході OC1 дорівнює $f_{OC1} = f_{clk_I/O}/2$ і досягається, якщо регістр OCR1A чи регістр ICR1 встановлюється в нуль (0x0000). В інших випадках частота сигналу на виході визначається за формулою

$$f_{OCn} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRn)},$$

де N є значенням коефіцієнта поділу попереднього подільника.

Режим швидкої широтно-імпульсної модуляції

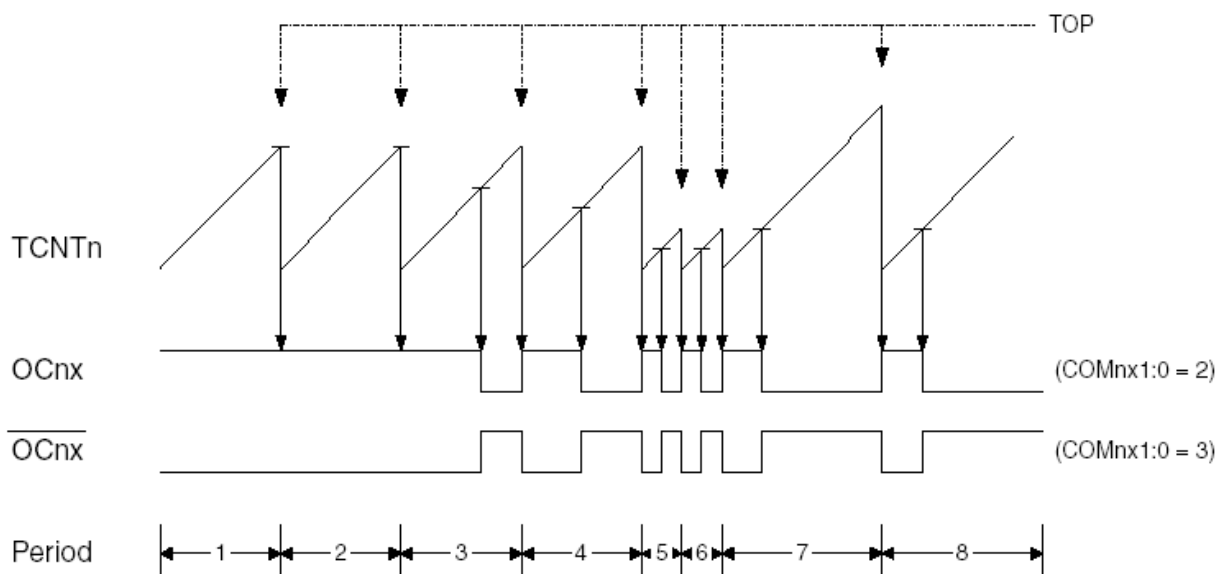


Рис. 153. Режим швидкої широтно-імпульсної модуляції

У режимі швидкої широтно-імпульсної модуляції (Fast PWM) лічильник рахує від 0 до максимального значення, яке залежить від режиму роботи. Якщо значення у лічильнику перевищує задане у регістрі OCR1, здійснюється перемикавання відповідного виходу. Вихід повертається у попередній стан при скиданні лічильника. При кожному скиданні лічильника відбувається

встановлення ознаки TOV0, а при досягненні значення OCR1A – ознаки OC1. У даному режимі роботи центр імпульсу на виході "плаває" відносно початку відліку.

У режимі швидкої широтно-імпульсної модуляції лічильник збільшується або до одного із фіксованих значень (0x00FF, 0x01FF, 0x03FF, 0xFFFF), або до значення, записаного у регістрі ICR1 чи регістрі OCR1A. Часова діаграма для режиму швидкої ШІМ показана на рис. 153, де зображено режим із використанням максимального значення з регістра OCR1A чи ICR1 (позначеного TOP). Значення TCNT1 на діаграмі показане у вигляді графіка. На часовій діаграмі зображено й неінвертований, і інвертований режими ШІМ.

Значення частоти вихідного сигналу визначається за допомогою формули

$$f_{OCnхPWM} = \frac{f_{clk_I/O}}{N \cdot (1 + TOP)},$$

де N є значенням коефіцієнта поділу попереднього подільника, а TOP – значення, до якого рахує лічильник, що може бути як фіксованим, так і залежати від вмісту регістра OCR1A чи регістра ICR1.

Режим широтно-імпульсної модуляції з коректною фазою

У режимі широтно-імпульсної модуляції із коректною фазою (PWM, Phase Correct) лічильник рахує від 0 до максимального значення, а потім у зворотний бік – від максимального значення до 0. Якщо значення у лічильнику перевищує задане у регістрі OCR1A, то здійснюється перемикання відповідного виходу. Коли в процесі зворотного відліку значення у лічильнику стане меншим, ніж значення, задане у OCR1A, то вихід повернеться у початковий стан. У даному режимі роботи положення центру імпульса на виході відносно початку відліку не залежить від вмісту регістра OCR1A. Частота зміни сигналу на виході у цьому режимі роботи є меншою, ніж у режимі швидкої широтно-імпульсної модуляції, проте, завдяки симетрії, цей метод має свої переваги. Часова діаграма для режиму широтно-імпульсної модуляції із коректною фазою зображено на рис. 154. Значення TCNT1 показане у вигляді графіка. На діаграмі також наведені неінвертований та інвертований виходи ШІМ.

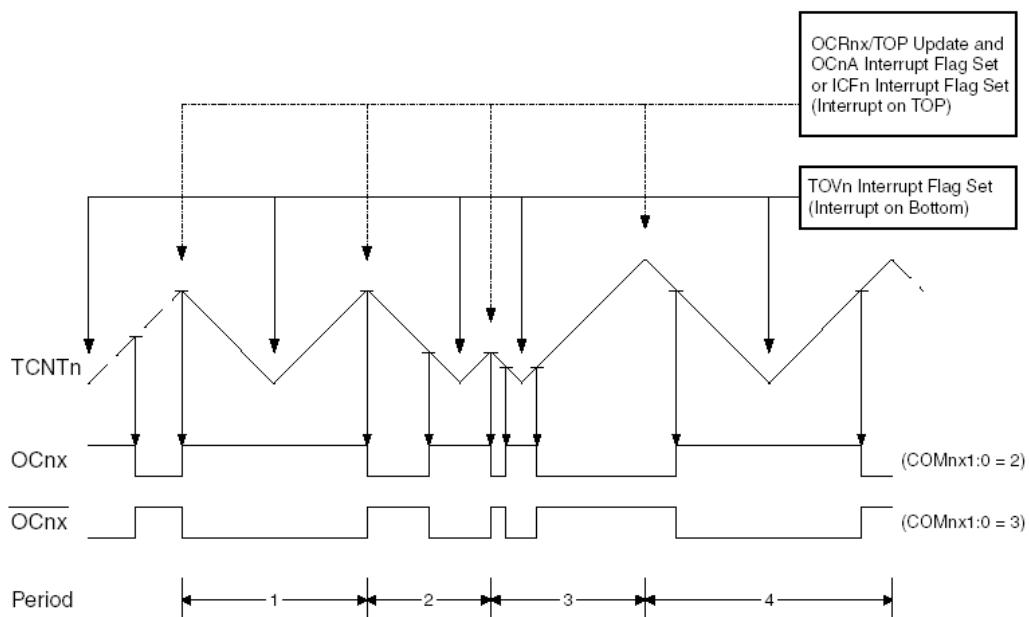


Рис. 154. Режим широтно-імпульсної модуляції з коректною фазою

Коли біти OCR1A чи ICR1 використовуються для визначення максимального значення, біти OC1A чи, відповідно, ICF1 встановлюється в 1 при досягненні лічильником максимального значення. Одночасно, якщо це потрібно, у регістрі, що використовується для задавання максимального значення, змінюється значення на нове.

Частота вихідного сигналу визначається за формулою

$$f_{OCnxPCPWM} = \frac{f_{clk_I/O}}{2 \cdot N \cdot TOP} ,$$

де N – значення коефіцієнта поділу попереднього подільника, а TOP – максимальне значення, до якого може дорахувати лічильник.

Режим широтно-імпульсної модуляції коректний по фазі та частоті

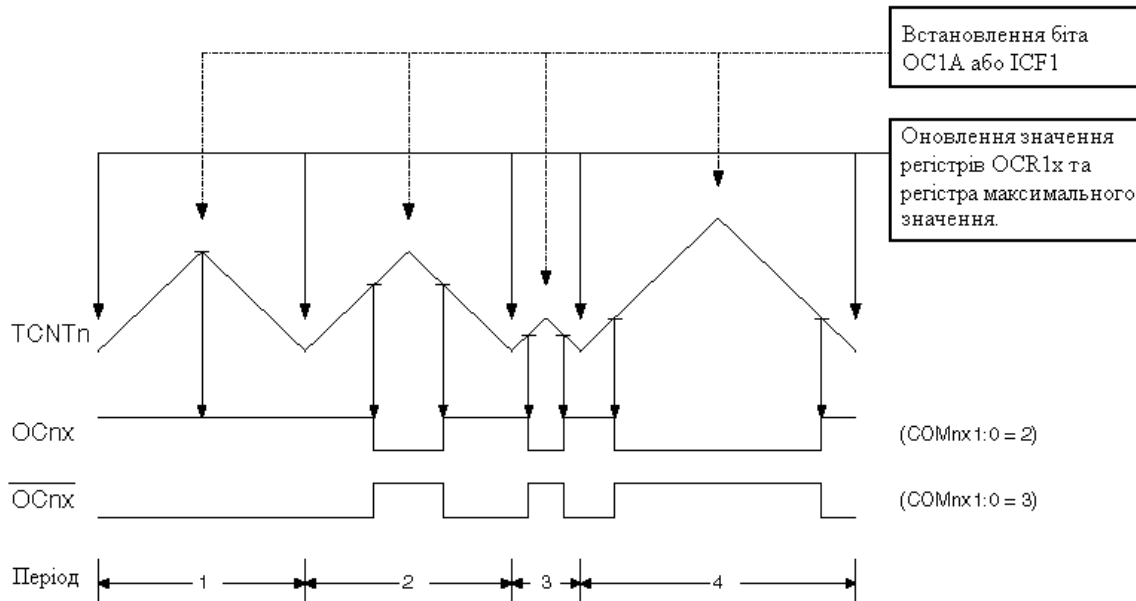


Рис. 155. Режим широтно-імпульсної модуляції коректний по фазі та частоті

Режим широтно-імпульсної модуляції, коректний по фазі та частоті, який подібний на режим із коректною фазою, базується на відліку лічильника у двох напрямках. Основна відмінність між режимами полягає у часі оновлення максимального значення, до якого може рахувати лічильник. У режимі з коректною фазою регістр, що задає максимальне значення відліку, змінюється у будь-який момент часу. У режимі з коректною фазою та частотою значення у регістрі максимального значення відліку змінюється лише у момент, коли лічильник досягнув мінімального значення і почав рахувати у бік збільшення. Часова діаграма режиму наведена на рис. 155.

Вихід лічильника-таймера 1

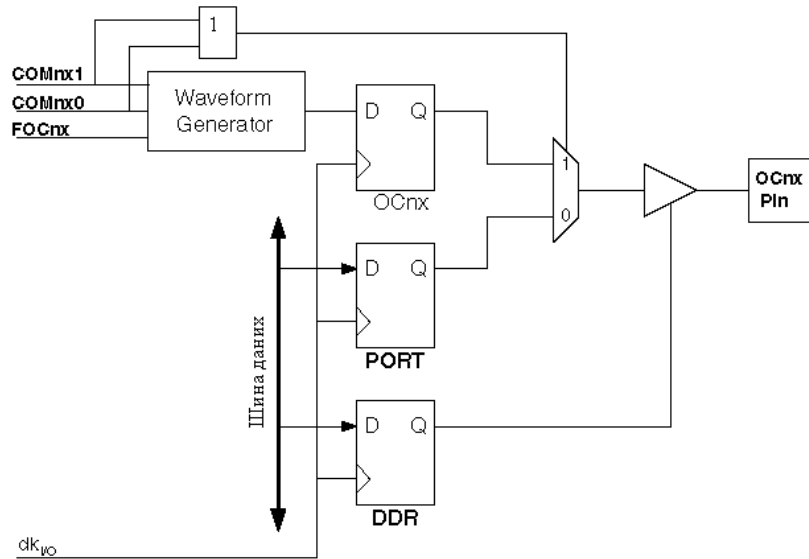


Рис. 156. Спрощена схема виходів таймера-лічильника 1

Таймер-лічильник 1 має два незалежні виходи, виведені на лінії OC1A та OC1B. Стан виходу залежить від обраного режиму роботи таймера-лічильника та конфігурації виходів. Конфігурація виходів задається для кожної лінії окремо бітами COM1A1:0 та COM1B1:0.

Спрощена схемотехніка виходу таймера наведена на рисунку 156. З неї бачимо, що видачею даних керує регістр напрямку DDR і для передавання даних необхідно, щоб лінія працювала на вивід. Якщо ж лінія буде працювати на ввід даних, стан виходу не буде перемикатися. При під'єднанні до лінії порту виходу лічильника, стан лінії може контролюватися за допомогою відповідного регістра PINx.

Режим роботи виходів OC1A та OC1B залежить від стану бітів COM1A1-0 та COM1B1-0 регістра TCCR1A, а також обраного режиму роботи.

Таблиця 72

Режими роботи ліній OC1A та OC1B залежно від стану бітів регістра TCCR1A у режимах, не пов'язаних із ШІМ

COM1x1	COM1x0	Опис роботи лінії OC1x
0	0	Нормальний режим роботи порту, біт OC1x від'єднаний від порту
0	1	Лінія OC1x перемикається при кожному співпаданні
1	0	Лінія OC1x скидається при кожному співпаданні
1	1	Лінія OC1x встановлюється при кожному співпаданні

Таблиця 73

Режими роботи ліній OC1x залежно від стану бітів регістра TCCR1A у режимі швидкого ШІМ

COM1x1	COM1x0	Опис роботи лінії OC1x
0	0	Нормальний режим роботи порту, біти OC1x від'єднані від порту
0	1	Якщо WGM13:0 = 1111, вихід OC1A перемикається при кожному співпаданні, а OC1B вимикається. В усіх решта режимах режим вважається зарезервованим, а біти OC1x від'єднані від порту
1	0	Біт OC1x скидається при рівності вмісту таймера (TCNT1) та регістра OCR1x, встановлюється при переході лічильника TCNT1 до нульового значення (неінвертуючий режим)
1	1	Біт OC1x встановлюється при рівності вмісту таймера (TCNT1) та регістра OCR1x, скидається при переході лічильника TCNT1 до нульового значення (інвертуючий режим)

Таблиця 74

Режими роботи ліній OC1x залежно від стану бітів регістра TCCR1A у режимах, коректних за фазою та режимі, коректному за фазою та частотою ШІМ

COM1x1	COM1x0	Опис роботи лінії OC1x
0	0	Нормальний режим роботи порту, біти OC1x від'єднані від порту.
0	1	Якщо WGM13:0 = 1111 вихід OC1A перемикається при кожному співпаданні, а OC1B вимикається. В усіх решта режимах режим вважається зарезервованим, а біти OC1x від'єднані від порту.
1	0	Біт OC1x скидається при рівності вмісту таймера (TCNT1) та регістра OCR1x, встановлюється при переході лічильника TCNT1 до нульового значення (неінвертуючий режим)
1	1	Біт OC1x встановлюється при рівності вмісту таймера (TCNT1) та регістра OCR1x, скидається при переході лічильника TCNT1 до нульового значення (інвертуючий режим)

Доступ до 16-бітних регістрів таймера-лічильника 1

Регістри TCNT1, OCR1A/B та ICR1, пов'язані з таймером-лічильником 1, є 16-бітними. Для доступу до цих регістрів, зазвичай, використовують дві,

наступні одна за одною операції читання чи запису. При звертанні до регістрів 16-розрядний таймер має один 8-розрядний регістр для тимчасового зберігання старшого байту, що є загальним для всіх 16-розрядних регістрів. При читанні молодшого байту із 16-бітних регістрів TCNT1, ICR1 старший байт автоматично розміщується у цей тимчасовий регістр і зчитується з нього, коли відбувається читання із регістра старшого байту. Так як буферний регістр є єдиним для всіх регістрів, немає можливості зчитати молодші байти із обох 16-бітних регістрів (TCNT1L, ICR1L), а потім звернутися до регістрів, що зберігають старші байти (TCNT1H, ICR1H, відповідно).

Запис даних у 8-бітні регістри TCNT1H, ICR1H, OCR1A, OCR1B не призводить до моментального оновлення даних. Дані лише розміщуються у тимчасовий регістр, перенесення даних у регістр старшого байту відбувається у момент запису даних у регістр молодшого байту. Тобто оновлюються усі 16 бітів і молодший байт задається командою, а старший – тимчасовим регістром.

Отже, для коректного звертання до регістрів TCNT1, ICR1 їх читання слід починати із молодшого байту, а запис – із старшого. Для регістрів OCR1A та OCR1B порядок читання байтів не важливий, а запис має починатися із старшого байту. При цьому не можна чергувати звертання до регістрів із різними назвами, а запис з читанням (крім читання регістрів OCR1A, OCR1B). Якщо у програмі підпрограми обробки переривань звертаються до 16-бітних регістрів таймера, слід унеможливити доступ до них із підпрограми обробки, якщо основна програма звернулася до одного з них. У найпростішому випадку, при звертанні до регістрів 16-бітних регістрів, які користуються тимчасовим регістром, слід забороняти переривання.

Наступний приклад показує типовий варіант звертання до цих 16-бітних регістрів.

```
TIM16_WriteTCNT1:  
in r18,SREG ; Зберігаємо загальну ознаку дозволу  
переривань  
cli ; забороняємо переривання  
out TCNT1H,r17 ; Записуємо у регістр TCNT1 дані із  
регістрів R17:R16  
out TCNT1L,r16  
out SREG,r18 Відновлюємо загальну ознаку дозволу  
ret
```

6.10.3.2. Регістри таймера-лічильника 1

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 157. Формат регістра TCCR1A

Для задавання режиму роботи таймера-лічильника використовуються регістри TCCR1A та TCCR1B. Формат регістра TCCR1A наведений на рис. 157, а регістра TCCR1B – на рис. 158.

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 158. Формат регістра TCCR1B

Біти WGM13-10 задають режим роботи таймера-лічильника (таблиця 71).

Біти COM1A1-0, COM1B1-0 задають режим роботи виходів OC1A та OC1B.

Біти FOC1A та FOC1B дозволяють примусово встановити вихід у режимах, не пов'язаних із ШІМ. Запис у біт FOC1A чи FOC1B одиниці примусово перемикає вихід OC1A чи OC1B відповідно. Біт не зберігає свого значення і читається завжди як 0.

Біт ICNC1 вмикає цифровий фільтр на вході пристрою захоплення (рис. 148). Біт ICES1 визначає фронт, за котрим спрацьовує пристрій захоплення. Якщо біт має нульове значення, то використовується спадаючий (задній) фронт, одиничне значення вказує на зростаючий фронт.

Біти CS12-10 задають коефіцієнт поділу попереднього подільника.

Регістр відліку TCNT1

Bit	7	6	5	4	3	2	1	0	
	TCNT1[15:8]								TCNT1H
	TCNT1[7:0]								TCNT1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 159. Регістр відліку TCNT1

Основним регістром таймера-лічильника 1 є 16-бітний регістр відліку TCNT1, що поділений на 2 регістри TCNT1H та TCNT1L (рис. 159). Регістр TCNT1L доступний як для прямого читання, так і для запису, а регістр TCNT1H читається та записується через тимчасовий програмно прозорий регістр TEMP. При читанні регістра TCNT1L старший байт таймера-лічильника розташовується у тимчасовому регістрі, а потім може бути зчитаний при звертанні до регістра TCNT1H. При запису нові дані, що пишуться у TCNT1H, розміщуються у регістр TEMP, а потім перезаписуються в регістр лічильника у момент запису у регістр TCNT1L.

Отже, читання регістра TCNT1 має починатися з молодшого байту, а запис – зі старшого, при цьому між доступом до молодшого і старшого байтів TCNT1 не має вклинюватися доступ до інших 16-бітних регістрів таймера-лічильника.

Регістр захоплення

Регістр захоплення ICR1 призначений для фіксації поточного стану таймера-лічильника за зовнішньою подією. Доступ до регістра здійснюється через два 8-бітні регістри ICR1H та ICR1L. Формат регістрів зображено на рисунку 160.

Регістр ICR1L доступний як для прямого читання, так і для запису, а регістр ICR1H читається та записується через тимчасовий програмно прозорий

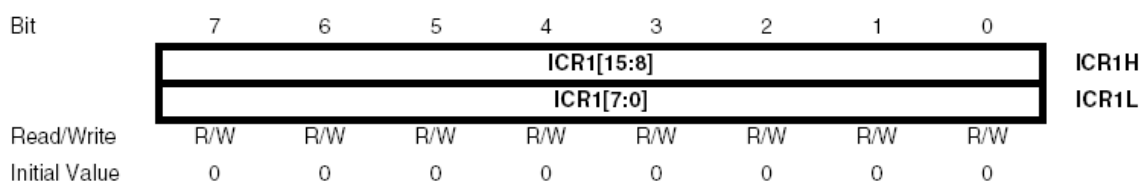


Рис. 160. Регістри захоплення ICR1H та ICR1L

регістр TEMP. При читанні регістра ICR1L старший байт таймера-лічильника розміщується у тимчасовий регістр, а потім може бути зчитаний при звертанні до регістра ICR1H. При запису нові дані, що пишуться у ICR1H, розміщуються в регістр TEMP, а потім перезаписуються у регістр лічильника в момент запису в регістр ICR1L. Отже, читання регістра ICR1 має починатись із молодшого

байту, а запис – зі старшого, при цьому між доступом до молодшого і старшого байтів ICR1 не має вклинюватися доступ до інших 16-бітних регістрів таймера-лічильника.

Регістри порівняння

Регістри порівняння призначені для збереження значення, з яким порівнюється значення лічильника. Результат порівняння може бути використаний для генерації переривань, широтно-імпульсної модуляції тощо. Таймер 1 має 2 регістри порівняння OCR1A та OCR1B. Довжина кожного регістра –16 бітів. Доступ до кожного з них здійснюється через два 8-бітні регістри. Для доступу до регістра OCR1A використовують OCR1AH та OCR1AL, а для доступу до OCR1B – регістри OCR1BH та OCR1BL.

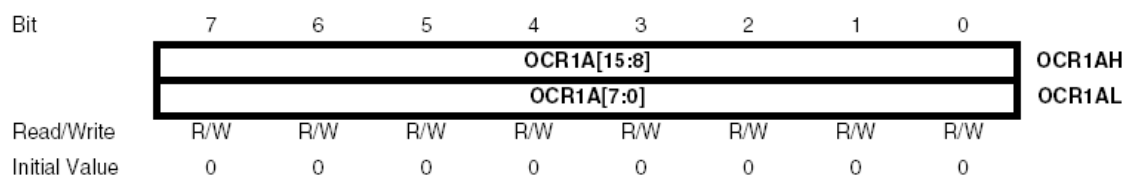


Рис. 161. Формат регістра OCR1A

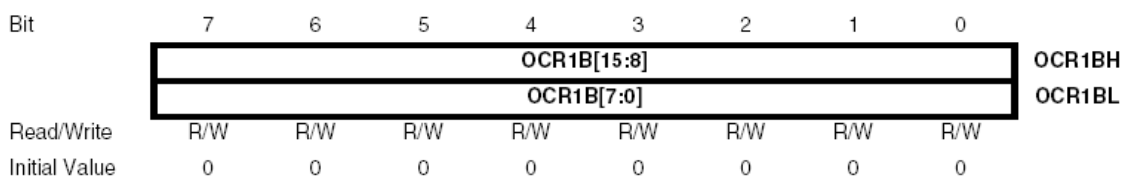


Рис. 162. Формат регістра OCR1B

Регістри OCR1AL та OCR1BL доступні для прямого читання та запису. У свою чергу, регістри OCR1AH та OCR1BH доступні для прямого читання, а записуються через тимчасовий програмно прозорий регістр TEMP: при записі нові дані, що пишуться у OCR1AH та OCR1BH, розміщуються у регістр TEMP, а потім перезаписуються у відповідний регістр у момент запису в регістр OCR1AL чи OCR1BL відповідно. Отже, для коректного запису даних запис в регістри OCR1A та OCR1B має починатися із старшого байту, при цьому між доступом до молодшого і старшого байтів регістра не повинен вклинюватися доступ до інших 16-бітних регістрів таймера-лічильника.

6.10.3.3. Біти загальних регістрів, що використовує таймер 1

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 163. Регістр маски переривань TIMSK

Регістр маски переривань TIMSK є загальним для усіх таймерів. Для таймера 0 використовуються біти OCIE0 та TOIE0.

Біт TICIE1, встановлений в 1, дозволяє переривання при захопленні для таймера-лічильника 1 (біт ICF1).

Біт OCIE1A, встановлений в 1, дозволяє переривання від блоку порівняння А таймера-лічильника 1 (біт OCF1A), а біт OCIE1B дозволяє переривання від блоку порівняння В (біт OCF1B).

Біт TOIE1, встановлений в 1, дозволяє переривання при переповненні таймера-лічильника 1 (біт TOV1).

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 164. Регістр переривань TIFR

У регістрі переривань таймерів TIFR відображається стан переривань від усіх таймерів мікросхеми. До таймера-лічильника 1 відносяться біти ICF1 OCF1A OCF1B TOV1.

Біт ICF1 вказує на захоплення значення лічильника у відповідному регістрі.

Біти OCF1A та OCF1B відображають стани запитів переривань від блоків порівняння А та В, а біт TOV1 – переривання при переповненні таймера. Біти скидаються автоматично при виклику відповідного переривання, крім того, вони можуть бути скинуті програмним шляхом.

6.10.4. Переривання від таймерів

Переривання, котрі можуть бути викликані таймерами-лічильниками, наведено у таблиці 75.

Переривання від таймерів-лічильників

Адреса переходу	Назва біту	Джерело виклику
\$008	OCF2	Співпадання в блоку порівняння таймера-лічильника 2
\$00A	TOV2	Переповнення таймера-лічильника 2
\$00C	ICF1	Відбулося захоплення стану у блоку захоплення таймера-лічильника 1
\$00E	OCF1A	Співпадання в блоку порівняння А таймера-лічильника 1
\$010	OCF1B	Співпадання в блоку порівняння В таймера-лічильника 1
\$012	TOV1	Переповнення таймера-лічильника 1
\$014	OCF0	Співпадання в блоку порівняння таймера-лічильника 0
\$016	TOVF0	Переповнення таймера-лічильника 0

Типова структура початкової області пам'яті програм наведена нижче.

```

.org 0
$000 jmp RESET           ;Вектор скидання
$002 jmp EXT_INT0       ;Вектор обробки IRQ0
$004 jmp EXT_INT1       ;Вектор обробки IRQ1
$006 jmp EXT_INT2       ;Вектор обробки IRQ2
$008 jmp TIM2_COMP      ;Вектор обробки переривання
                        ;співпадання в блоку
                        ;порівняння лічильника-
                        ;таймера 2
$00A jmp TIM2_OVF       ;Вектор обробки
                        ;переповнення
                        ;лічильника-таймера 2
$00C jmp TIM1_CAPT      ;Вектор обробки захоплення
                        ;стану у блок
                        ;захоплення лічильника-
                        ;таймера 1
$00E jmp TIM1_COMPA     ;Вектор обробки співпадання
                        ;в блоку порівняння А
                        ;лічильника-таймера 1
$010 jmp TIM1_COMPB     ;Вектор обробки співпадання
                        ;в блоку порівняння В
                        ;лічильника-таймера 1
$012 jmp TIM1_OVF       ;Вектор обробки
                        ;переповнення
                        ;лічильника-таймера 1
$014 jmp TIM0_COMP      ;Вектор обробки співпадання
                        ;в блоку порівняння
                        ;лічильника-таймера 0
$016 jmp TIM0_OVF       ;Вектор обробки переповнення

```



```

;лічильника-таймера 0
$018 jmp SPI_STC ;Вектор обробки закінчення
;передавання даних по шині SPI
$01A jmp USART_RXC ;Вектор обробки закінчення
;прийому USART
$01C jmp USART_UDRE ;Вектор обробки очищення
;регістра UDR
$01E jmp USART_TXC ;Вектор обробки закінчення
;передачі USART
$020 jmp ADC ;Вектор обробки закінчення
;аналого-цифрового перетворення
$022 jmp EE_RDY ;Вектор обробки готовності
;EEPROM
$024 jmp ANA_COMP ;Вектор обробки переривання від
;аналогового компаратора
$026 jmp TWI ;Вектор обробки переривання від
;шини TWI
$028 jmp SPM_RDY ;Вектор готовності пам'яті
;програм

```

6.11. Загальна характеристика послідовних інтерфейсів мікросхеми

Мікросхема ATМega32 у своїй структурі має три послідовних інтерфейси:

- ◇ синхронний послідовний інтерфейс SPI;
- ◇ асинхронний інтерфейс TWI (аналог інтерфейсу I2C);
- ◇ послідовний синхронно-асинхронний універсальний інтерфейс USART.

Кожен із інтерфейсів таймерів-лічильників орієнтований на виконання свого кола задач і має відповідні особливості.

Послідовний периферійний інтерфейс SPI (Serial Peripheral Interface) призначений для організації обміну даними між двома пристроями. З його допомогою може здійснюватися обмін даними між мікроконтролером і різними пристроями, такими, як цифрові потенціометри, ЦАП та АЦП, FLASH-ПЗП тощо. За допомогою цього інтерфейсу зручно проводити обмін даними між кількома мікроконтролерами.

Двопровідний послідовний інтерфейс TWI (Two-wire Serial Interface) є повним аналогом базової версії інтерфейсу I2C фірми Philips. Цей інтерфейс дозволяє об'єднати разом до 128 різних пристроїв за допомогою двобічної шини, що складається з лінії тактового сигналу (SCL) і лінії даних (SDA). Проте швидкість обміну за цим інтерфейсом є меншою, ніж за інтерфейсом SPI.

Універсальний асинхронний або універсальний синхронно/асинхронний приймач (Universal Synchronous / Asynchronous Receiver and Transmitter - UART або USART) – зручний і простий послідовний інтерфейс для організації інформаційного каналу обміну мікроконтролера із "зовнішнім світом". Він здатний працювати у дуплексному режимі (одночасне передавання і приймання даних) і підтримує протокол стандарту RS-232, що забезпечує можливість організації зв'язку з персональним комп'ютером.

6.11.1. Послідовний порт SPI

6.11.1.1. Загальний опис логіки роботи

Інтерфейс SPI (Serial Peripheral Interface, SPI bus — послідовний периферійний інтерфейс,) — послідовний синхронний стандарт передавання даних у режимі повного дуплексу, розроблений компанією Motorola для забезпечення простого та дешевого спряження мікроконтролерів та периферії. В інтерфейсі SPI є два типи пристроїв: ведучий, котрий керує передаванням, та ведений, котрий приймає та передає дані за командами ведучого. Зазвичай, на шині SPI є один ведучий пристрій та один чи кілька ведених.

На відміну від більшості інших, інтерфейс SPI є синхронним, у котрому будь-яке передавання даних синхронізоване зі спільним тактовим сигналом, що генерується або ведучим пристроєм, або незалежним тактовим генератором. Ведені мікросхеми синхронізують отримання бітової послідовності з тактовим сигналом. Ведучий пристрій (чи один із ведучих) обирає ведений пристрій за допомогою сигналу SS (slave select). Мікросхеми, не обрані сигналом SS, не мають права видавати дані на шину (хоча й можуть приймати дані). У шині SPI використовують 4 цифрові сигнали:

- ◇ MOSI (або SI) – вихід ведучого, вхід веденого (Master Out Slave In), використовується для передавання даних від ведучого пристрою до веденого.
- ◇ MISO (або SO) – вхід ведучого, вихід веденого (Master In Slave Out), використовується для передавання даних від веденого пристрою до ведучого.
- ◇ SCK (або CLK) – послідовний тактовий сигнал (Serial Clock), використовується для передавання тактового сигналу для ведених пристроїв та, можливо, для ведучих.
- ◇ SS (або CS) – вибір мікросхеми (Chip Select, Slave Select) для ведених пристроїв - вибір активного веденого пристрою, для деяких ведучих пристроїв може використовуватися для арбітражу.

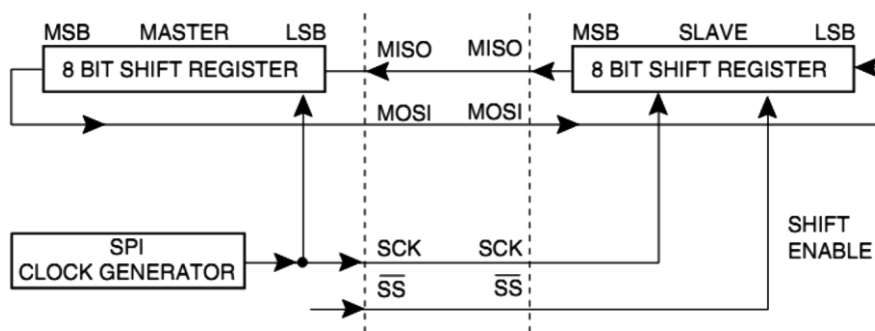


Рис. 165. Зв'язок за інтерфейсом SPI

Типова структура апаратних засобів може бути представлена за допомогою двох регістрів зсуву, що утворюють 16-бітний кільцевий регістр зсуву. Передавання даних синхронізується тактовим сигналом, котрий може бути як зовнішнім, так і внутрішнім для ведучого пристрою.

Перед початком передачі ведучий пристрій передає на лінію SS веденої мікросхеми логічний 0. Зауважимо, що деякі ведені пристрої потребують певної затримки між подаванням сигналу вибірки та подаванням на лінію SCK тактових сигналів.

Далі, на кожному із тактів сигналу SCK, ведучий пристрій посилає на лінію MOSI новий біт із зсувного регістра, а з лінії MISO читає біт даних, надісланий веденим пристроєм, і розміщує їх у зсувний регістр. Ведений

пристрій, у свою чергу, в той самий час посилає на лінію MISO нові біти даних із свого зсувного регістра, а з лінії MOSI читає біти від ведучого пристрою. Отже, через 8 тактів ведений та ведучий пристрої обмінюються байтами даних. Не завжди потрібно проводити передавання даних у двох напрямках – у такому випадку не потрібні дані можна просто ігнорувати, а лінію, за котрою буде відбуватися не потрібне передавання даних, можна взагалі не під'єднувати до веденої мікросхеми. У багатьох випадках передачі можуть містити будь-яку кількість бітів, але найчастіше число бітів, що передається, є кратним 8.

На додаток до тактової частоти, у процесі програмування слід також обрати полярність тактового сигналу та фазу даних відносно тактового сигналу. Звичайно вибір полярності тактового сигналу та фази даних виконується бітами CPOL і CPHA у регістрі керування. Біт CPOL задає базове значення тактового сигналу.

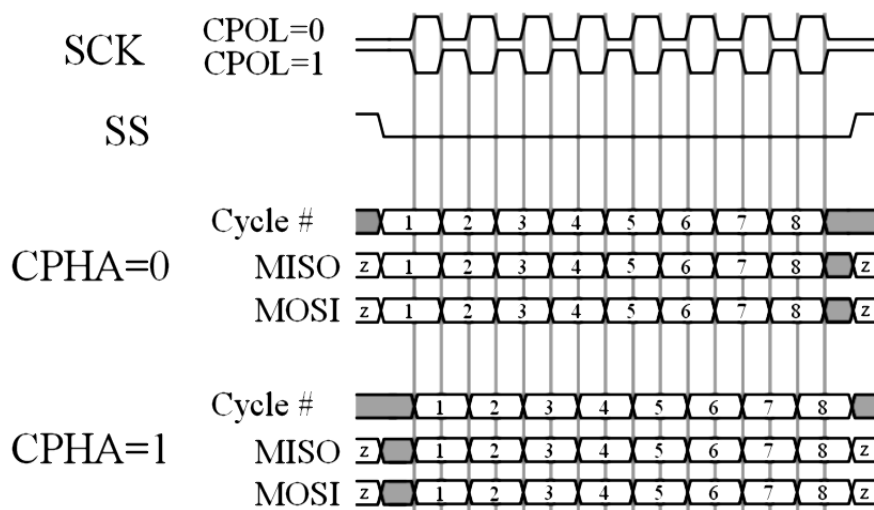


Рис. 166. Полярність синхронізації та фаза даних, що передаються

Значення рівнів сигналів SCK та CPHA

CPOL	Активний рівень сигналу SCK	CPHA	Фаза даних
0	Логічний 1	0	На наявність даних вказує перехід сигналу SCK у активний рівень
1	Логічний 0	1	На наявність даних вказує перехід сигналу SCK у пасивний рівень

6.11.1.2. Конфігурації з багатьма веденими

При роботі порту у веденому режимі (Slave) інтерфейс SPI залишається вимкненим, а його передавач у z-стані, поки на лінії SS високий рівень. У цьому стані програма може оновлювати вміст регістра даних SPI, але дані не будуть зсунуті до встановлення на лінії SS низького рівня навіть за наявності тактового сигналу. Така організація дозволяє легко під'єднувати до шини SPI кілька ведених мікросхем. Найширше використовують увімкнення із незалежним використанням мікросхем. У такій схемі кожна ведена мікросхема вибирається власним сигналом SS (рис. 167).

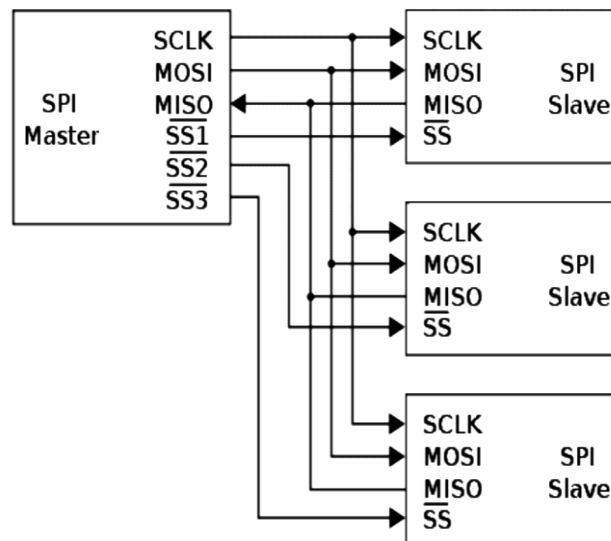


Рис. 167. Незалежне увімкнення ведених мікросхем

6.11.1.3. Інтерфейс SPI у мікросхемі ATmega32

У мікросхемах сімейства ATmega інтерфейс SPI має такі можливості:

- ◇ Повний дуплекс при трипровідному синхронному передаванні даних.

- ◇ Використання мікросхеми як у режимі ведучого, так і у режимі веденого пристрою.
- ◇ Можливість передавання даних, починаючи як зі старшого, так і з молодшого біту.
- ◇ Можливість програмування швидкості робом.

У сімействі ATmega інтерфейс SPI має однократну буферизацію при передаванні даних й подвійну буферизацію при прийомі: байти, що повинні передаватися, не можуть бути записані в регістр даних SPI до закінчення циклу попереднього передавання даних. При прийомі, однак, отримані дані можуть бути прочитані з регістру даних SPI до кінця наступного приймання даних.

При дозволеному SPI напрямі передавання даних лінії MOSI, MISO, SCK і SS налаштовуються у відповідності з наступною таблицею.

Таблиця 77

Налаштування виводів SPI

Вивід	Ведучий SPI	Ведений
MOSI	Визначається користувачем	Вхід
MISO	Вхід	Визначається користувачем
SCK	Визначається користувачем	Вхід
SS	Визначається користувачем	Вхід

Якщо SPI працює у режимі веденого (Slave), сигнал SS працює як вхід. Коли на лінії SS низький рівень, активізується порт SPI і лінії інтерфейсу працюють, як показано у таблиці 66. За наявності на лінії SS високого стану, усі виводи є входами, крім, можливо, лінії MISO, а SPI переводиться у пасивний стан, навіть якщо почалося нове передавання чи приймання даних. При переведенні лінії SS у високий рівень ведена мікросхема зупиняє передавання та приймання даних, та видаляє із регістра зсуву будь-які частково прийняті дані.

У режимі ведучого SPI напрям роботи лінії SS задається. Якщо лінія SS налаштована як вихід, стан лінії SS не впливає на роботу SPI. Якщо ж лінія SS

налаштована як вхід, її стан дозволяє провести визначення активної ведучої мікросхеми у системах із кількома ведучими контролерами: високий рівень на лінії SS дозволяє роботу мікросхеми у режимі ведучого, а низький рівень автоматично переводить мікросхему у режим веденої.

Будова послідовного порту SPI зображена на рис. 168. Для отримання тактового сигналу синхронізації використовується тактовий сигнал XTAL, який надходить на подільник DIVIDER, коефіцієнт поділу котрого визначається мультиплексором SELECT. Отриманий тактовий сигнал пнадходить на логічний елемент (CLOCK LOGIK) та через нього – на зсувний регістр (SHIFT REGISTER). Вихідна логіка узгоджує сигнали із зсувного регістра із заданим режимом роботи ліній передавання даних. Задавання режиму роботи та керування блоками послідовного порту здійснює блок керування (SPI CONTROL), а встановлення режиму роботи – регістр режиму (SPI CONTROL REGISTER). Стан послідовного порту контролюється через регістр стану (SPI STATUS REGISTER).

Для контролю стану, програмування, передавання та приймання даних використовують такі регістри:

- ◇ SPDR – регістр даних SPI (SPI Data Register);
- ◇ SPCR – регістр керування SPI (SPI Control Register);
- ◇ SPSR – регістр стану SPI (SPI Status Register).

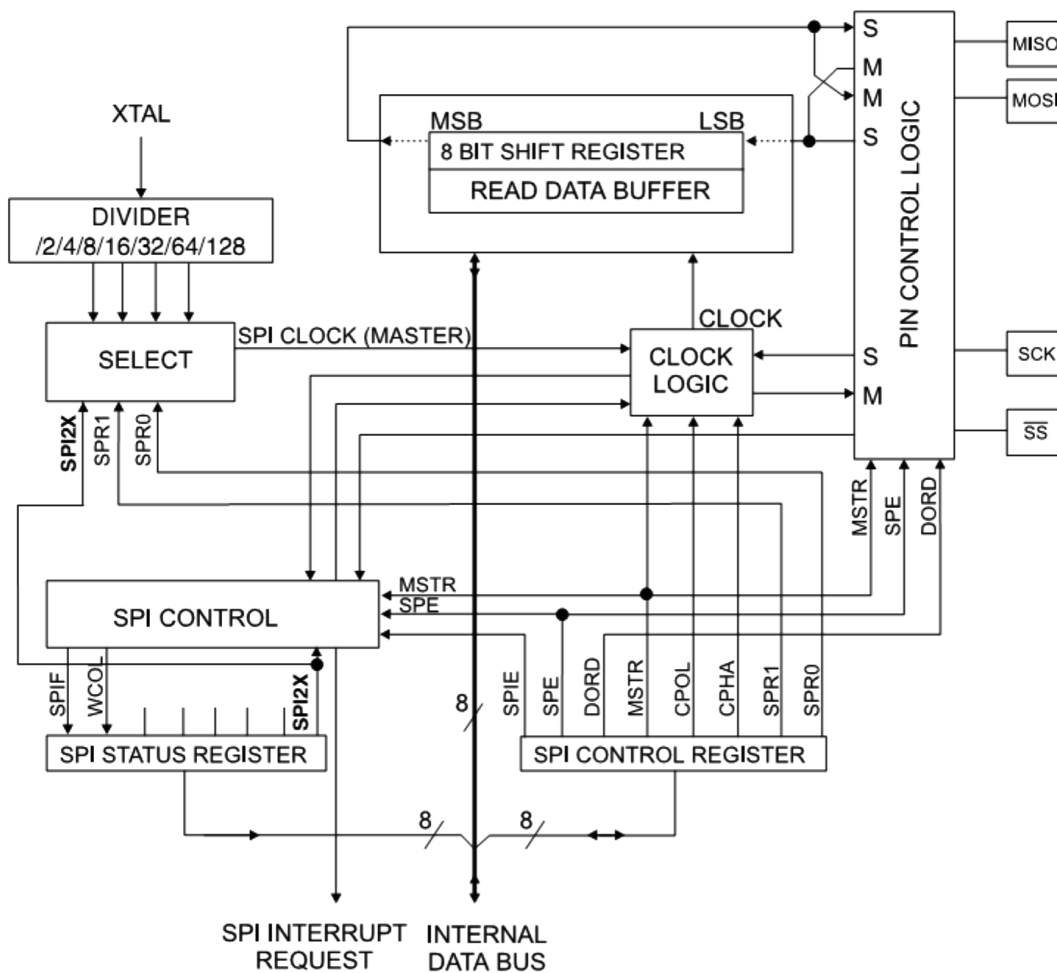


Рис. 168. Будова послідовного порту

6.11.1.4. Регістри SPI

Регістр SPDR призначений для приймання та передавання даних через SPI. Дані, що передаються, не буферизуються, а для прийнятих даних є один однобайтовий буфер. Так як дані, що передаються, не буферизуються, перед записом нових даних у регістр SPDR слід дочекатися закінчення попередньої передачі.

Формат регістра SPCR зображений на рисунку 169.

Bit	7	6	5	4	3	2	1	0	
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 169. Регістр SPCR

Біти цього регістра мають таке призначення:

SPIE – (SPI Interrupt Enable) дозвіл переривання від SPI. Біт дозволяє генерацію переривання від інтерфейсу SPI при встановленні біту SPIF у регістрі SPSR.

SPE – (SPI Enable) дозвіл роботи інтерфейсу SPI. Одиничне значення біту дозволяє роботу інтерфейсу SPE, нульове – забороняє її.

DORD – (Data Order) порядок передавання даних. Одиниця вказує, що дані передаються від молодших бітів до старших (від розряду D0 до D7), нуль вказує, що дані передаються від старших бітів до молодших (від D7 до D0).

MSTR – (Master/Slave Select) біт вибору режиму роботи «ведучий» чи «ведений». Якщо біт дорівнює 1, то порт працює у ведучому режимі, якщо нуль – у веденому.

CPOL – (Clock Polarity) біт вибору полярності сигналу SCK. Якщо біт дорівнює 1, то активний рівень тактового сигналу – нульовий, а пасивний – одиничний. Якщо нуль – активний рівень сигналу одиничний, а пасивний – нульовий.

Таблиця 78

Частота передавання даних

SPI2X	SPR1	SPR0	Частота сигналу SCK та передавання даних
0	0	0	fosc/4
0	0	1	fosc/16
0	1	0	fosc/64
0	1	1	fosc/128
1	0	0	fosc/2
1	0	1	fosc/8
1	1	0	fosc/32
1	1	1	fosc/64

fosc – частота сигналу генератора

CPHA – (Clock Phase) біт вибору фази передавання даних. Нульове значення вказує на наявність даних при переході сигналу SCK у активний рівень, одиничне – на наявність даних при переході сигналу SCK у пасивний рівень.

SPR1, SPR0 – (SPI Clock Rate Select) – біти вибору частоти передавання даних. Біти разом із бітом SPI2X визначають швидкість передавання даних по SPI у режимі ведучої мікросхеми із внутрішнім тактовим сигналом. У режимі веденої мікро-схеми чи у режимі ведучої із зовнішнім тактовим сигналом біти не використовуються. Частота тактового сигналу визначається згідно з таблицею 78.

Біт SPI2X знаходиться у регістрі SPSR.

Формат регістра SPSR зображений на рисунку 170.

Bit	7	6	5	4	3	2	1	0	
	SPIF	WCOL	-	-	-	-	-	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 170. Формат регістра SPSR

Біт SPIF (SPI Interrupt Flag) вказує на переривання від порту SPI. Біт встановлюється в 1 при закінченні обміну даними, а у режимі ведучого порту також при передаванні даних керування шиною іншому контролеру (тобто тоді, коли на вхід SS надійшов нуль). При встановленні біту в 1 генерується переривання, якщо біт SPIE у регістрі SPCR встановлений в 1 та встановлений біт загального дозволу переривань. Біт автоматично скидається при виклику відповідного переривання та при виконанні операції читання регістра SPSR.

Біт WCOL (Write COLLision Flag) – біт колізії при запису. Одиничне значення вказує на записування у регістр даних SPDR під час передавання даних. Біт автоматично скидається при виконанні операції читання регістра SPSR.

Біт SPI2X (Double SPI Speed Bit) – біт подвоєння швидкості передавання даних. Біт обирає швидкість передачі даних по SPI разом із бітами SPR1, SPR0 (таблиця 67).

6.11.2. Розширення портів вводу-виводу

Іноді, при проектуванні пристроїв виникає ситуація, коли виводів мікроконтролера не вистачає, а використати інший мікроконтролер немає можливості. У таких випадках використовують розширення за допомогою зовнішніх розширювачів вводу-виводу, що під'єднуються до мікроконтролера за стандартним інтерфейсом, наприклад, інтерфейсом SPI чи I²C(TWI). Найпростіший варіант такого розширення – використання зсувних регістрів чи мультиплексорів, однак у такому випадку лінії портів, що розширюються, будуть працювати виключно на вхід чи на вихід без можливості програмного налаштування напрямку передавання даних, але таке налаштування у більшості випадків і не потрібне.

Для вводу даних у мікроконтролер може бути використаний взагалі будь-який зсувний регістр, що має паралельні входи завантаження і послідовний вихід даних. На схемі, зображеній на рис. 171, розширення вводу здійснюється за допомогою зсувного регістра типу 74НТС165 (К555ІР9). Перед зчитуванням даних на програмно керовану лінію \overline{CAP} встановлюється логічний 0, що призводить до запису даних із паралельних входів D15 - D0 у зсувний регістр. Потім лінія \overline{CAP} встановлюється у стан логічної 1 і регістр перемикається у режим зсуву. Коли послідовним інтерфейсом SPI будуть прийняті 2 байти, зчитування даних буде завершено і регістри можна знову перевести у режим паралельного завантаження. В принципі, у якості сигналу \overline{CAP} можна використовувати проінвертований сигнал SS (PB4). Позначення D15 - D0 відповідають позиціям бітів у двобайтовому слові, що буде прийнято інтерфейсом SPI.

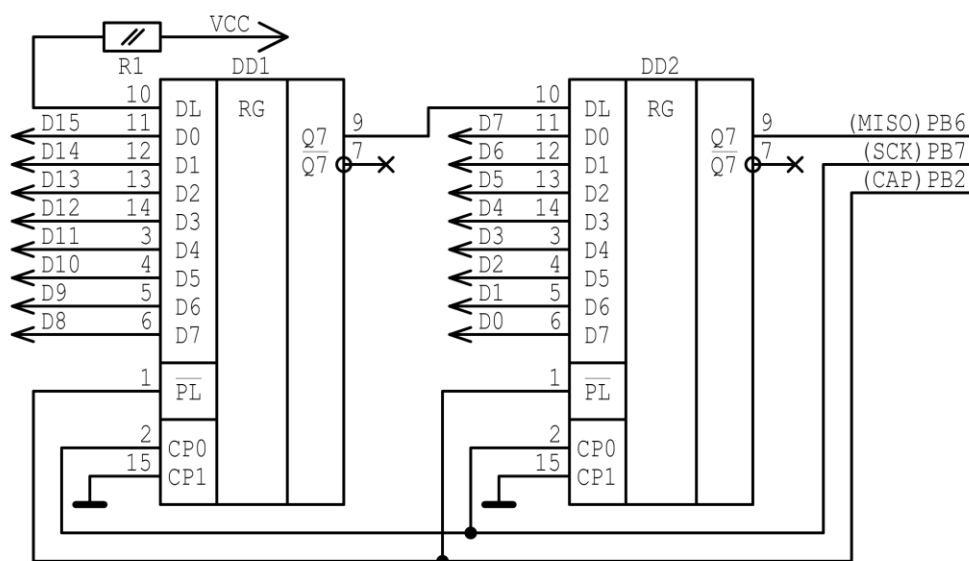


Рис. 171. Розширення портів вводу за допомогою інтерфейсу SPI. Варіант послідовного читання байтів

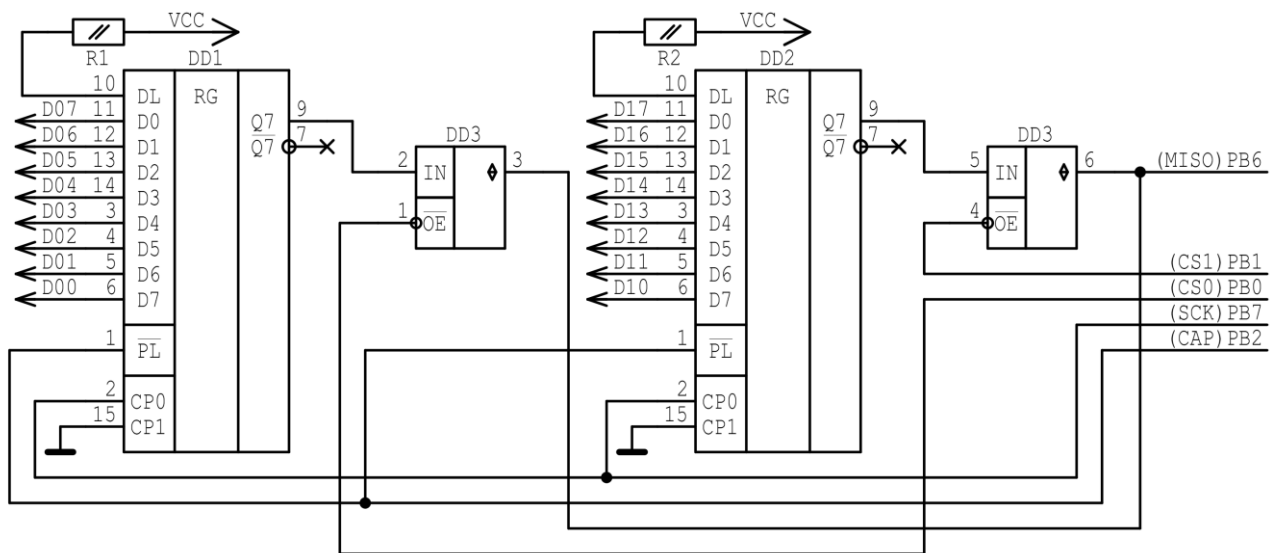


Рис. 172. Розширення портів вводу за допомогою інтерфейсу SPI. Варіант із вибором регістра окремим сигналом

Інший варіант схеми (рис. 172) може мати сигнали вибору, за котрими вихід послідовного коду потрібного зсувного регістра під'єднується до лінії MISO, а не вибрані регістри - вимикаються від неї. Така схема не буде потребувати зчитування усіх байтів для визначення стану одного конкретного порту, але є складнішою, містить більшу кількість мікросхем і потребує додатковий сигнал CS_i на кожен додатковий порт вводу. У якості буфера із Z станом (DD3) можна використати 74LS125 (K555АП8). Вибір джерела сигналу для SPI також можна здійснювати мультиплексором або за допомогою іншої мікросхеми.

Натомість, при розширенні порту виводу, використання звичайного зсувного регістра без буферизації може призвести до появи хибних імпульсів на створеному порті виводу. У деяких випадках поява короточасних імпульсів на виходах, наприклад, коли до виходів під'єднані індикатори, не є критичною і не призводить до хибного спрацювання механізмів чи обладнання. Якщо ж поява таких імпульсів не має бути дозволена, слід застосовувати зсувні регістри із буферизацією даних на паралельних виходах, наприклад, зсувний регістр 74HC595. Цим регістром можна керувати як за допомогою звичайних виводів мікроконтролера, так і за допомогою інтерфейсу SPI. Також його можна каскадувати, з'єднуючи мікросхеми у єдиний великий зсувний регістр.

Мікросхема 74HC595 є 8-розрядним зсувним регістром з регістром зберігання і вихідними буферами з трьома станами.

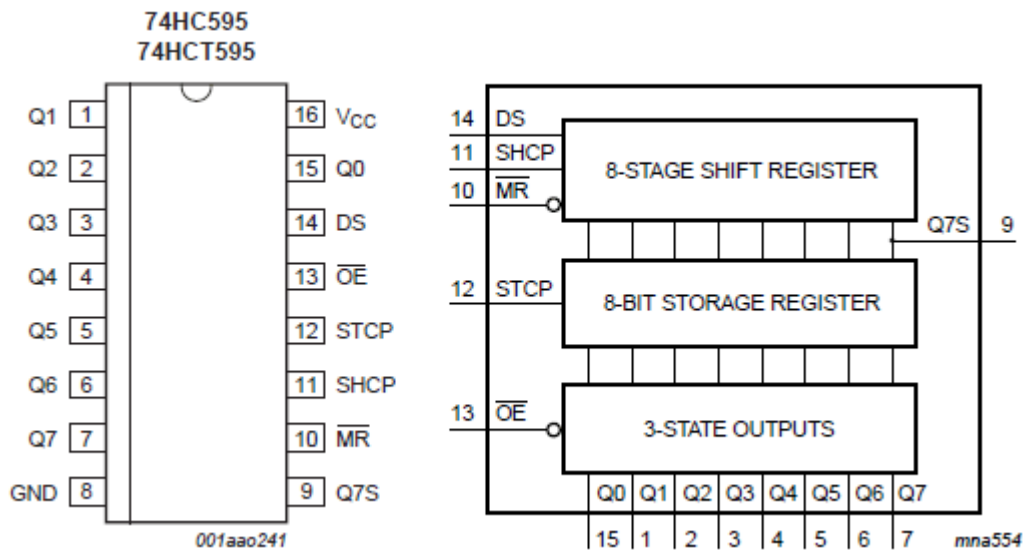


Рис. 173. Зсувний регістр 74HC595

Призначення її виводів є типовим для таких регістрів:

DS – вхід даних зсувного регістра у послідовному коді;

SHCP – тактовий вхід зсувного регістра;

MR – вхід скиду зсувного регістра;

STCP – тактовий вхід регістра зберігання даних;

OE – вхід дозволу вихідних буферів;

Q0...Q7 – виходи даних у паралельному коді;

Q7S – вихід каскадування;

GND, VCC – виводи живлення.

Вхід скидання MR у робочому стані підтягнутий до плюса живлення. Низький логічний рівень на цьому виводі стирає вміст зсувного регістра. Вміст регістра зберігання при цьому ніяк не змінюється.

На вході DS встановлюється необхідний логічний рівень. За позитивним перепадом тактового сигналу на вході SHCP вміст зсувного регістра (розряди з 0 по 7) зміщується на один розряд, а нульовий розряд регістра зберігає логічний рівень, встановлений на вході DS. При зміщенні сьомий розряд зсувного регістра не затирається, а зберігається у внутрішньому тригері й транслюється на вивід Q7S, який призначений для каскадування зсувного регістра.

Для запису одного байта даних описану послідовність потрібно повторити вісім разів.

За позитивним перепадом тактового сигналу на вході STCP дані з виходу зсувного регістра записуються в регістр зберігання. Якщо на виводі OE буде низький логічний рівень, то дані регістра зберігання встановляться на виходах Q0 ... Q7, в іншому випадку ці виходи будуть знаходитися в третьому стані.

Для каскадування зсувного регістра вихід Q7S під'єднують до входу даних DS наступного регістра, а виводи SHCP, MR, STCP, OE одного регістра з'єднують з аналогічними виводами іншого. Приклад каскадного ввімкнення двох зсувних регістрів можна бачити на схемі нижче.

Управління зсувними регістром 74HC595 можна реалізувати як програмно, так і апаратно, використовуючи SPI модуль мікроконтролера AVR. Останнє можливе завдяки тому, що часова діаграма сигналів керування 74HC595 збігається з діаграмою формованої модулем SPI.

У разі програмного керування виводи зсувного регістра пі'єднуються до будь-яких виводів загального призначення. Для апаратного керування зсувними регістром його потрібно під'єднати таким чином:

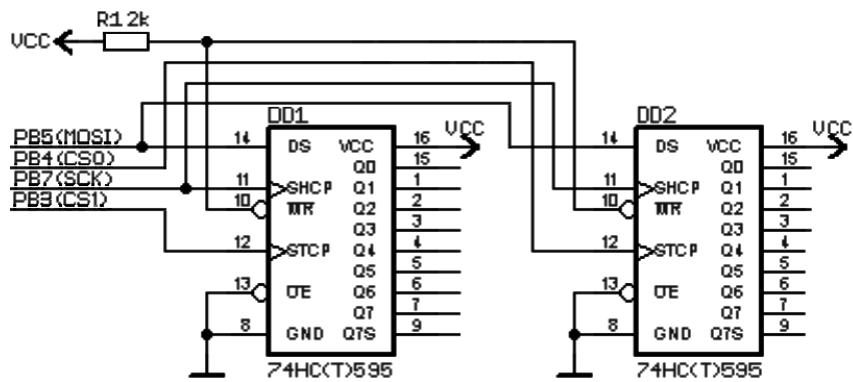
DS - > MOSI;

SHCP - > SCK;

MR - > через резистор до VCC;

OE - > GND.

Лінія STCP використовується для вибору потрібного регістра, що буде завантажуватися. Є два варіанти під'єднання: лінію можна використовувати для вибору активного регістра і здійснювати завантаження лише у потрібний регістр, а можна дані пересилати в усі регістри, з'єднавши у ланцюжок усі регістри каскадно й використовувати сигнал для вказання, що всі дані знаходяться на своїх місцях. У першому випадку кожен регістр може бути завантажений незалежно від інших, проте схема потребує сигналу CS на кожен регістр і не може змінювати стан усіх портів одночасно. Для запису даних у порт необхідно скинути сигнал CS_i відповідного порту, переслати 8 біт даних і встановити сигнал CS_i у високий рівень.

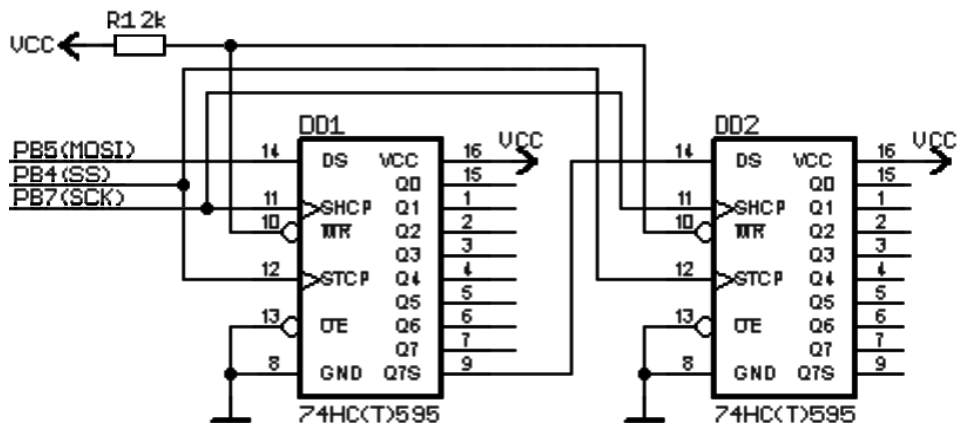


а)

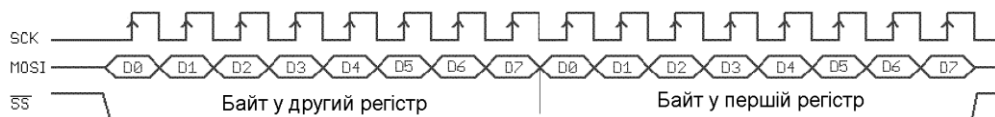


б)

Рис. 174. Розширення портів виводу за допомогою інтерфейсу SPI. Схема з незалежним вибором регістрів: а) – схема, б) – часова діаграма роботи



а)



б)

Рис. 175. Розширення портів виводу за допомогою інтерфейсу SPI. Схема з одночасним записом у регістри. а) – схема, б) – часова діаграма роботи

У другому випадку усі регістри треба перезавантажувати при зміні стану хоча б одного біта, але схема потребує меншу кількість ліній і змінює стан усіх вихідних портів одночасно. Для запису даних у порт необхідно скинути сигнал

SS, переслати всі дані для всіх портів, починаючи із останнього у ланцюжку, і встановити сигнал SS. В обох схемах виходи OE усіх регістрів під'єднанні до нуля, що “назавжди” дозволяє виходи регістрів, виводи MR підтягнуті через резистор до плюса живлення.

Усі наведені вище схеми можуть бути з легкістю розширені до необхідної кількості виводів.

6.11.3. Універсальний синхронно-асинхронний послідовний прийомопередавач USART

6.11.3.1. Загальний опис

Універсальний синхронно-асинхронний послідовний приймач (Universal Synchronous and Asynchronous serial Receiver and Transmitter — USART) є дуже гнучким пристроєм послідовної передачі інформації. Спрощена блок-схема передавача USART пнаведена на рис. 176.

Він має такі основні особливості:

- ◇ незалежні регістри послідовного приймання й передавання даних та можливість одночасного приймання та передавання даних;
- ◇ синхронний і асинхронний режими роботи;
- ◇ синхронізацію як від ведучого, так і від веденого пристрою;
- ◇ вибір швидкості передачі інформації в широких межах;
- ◇ підтримку кадрів довжиною 5–9 бітів і 1 або 2 стоп-біти;
- ◇ апаратну підтримку генерації і перевірки сигналу парності;
- ◇ виявлення переповнювання даних;
- ◇ виявлення помилок кодування;
- ◇ низькорівневу цифрову фільтрацію і виявлення помилкового стопового біту;
- ◇ три джерела переривання: "передача завершено", "регістр даних передавача порожній", "прийом завершено";
- ◇ режим міжпроцесорного зв'язку;
- ◇ два швидкісні режими асинхронної передачі.

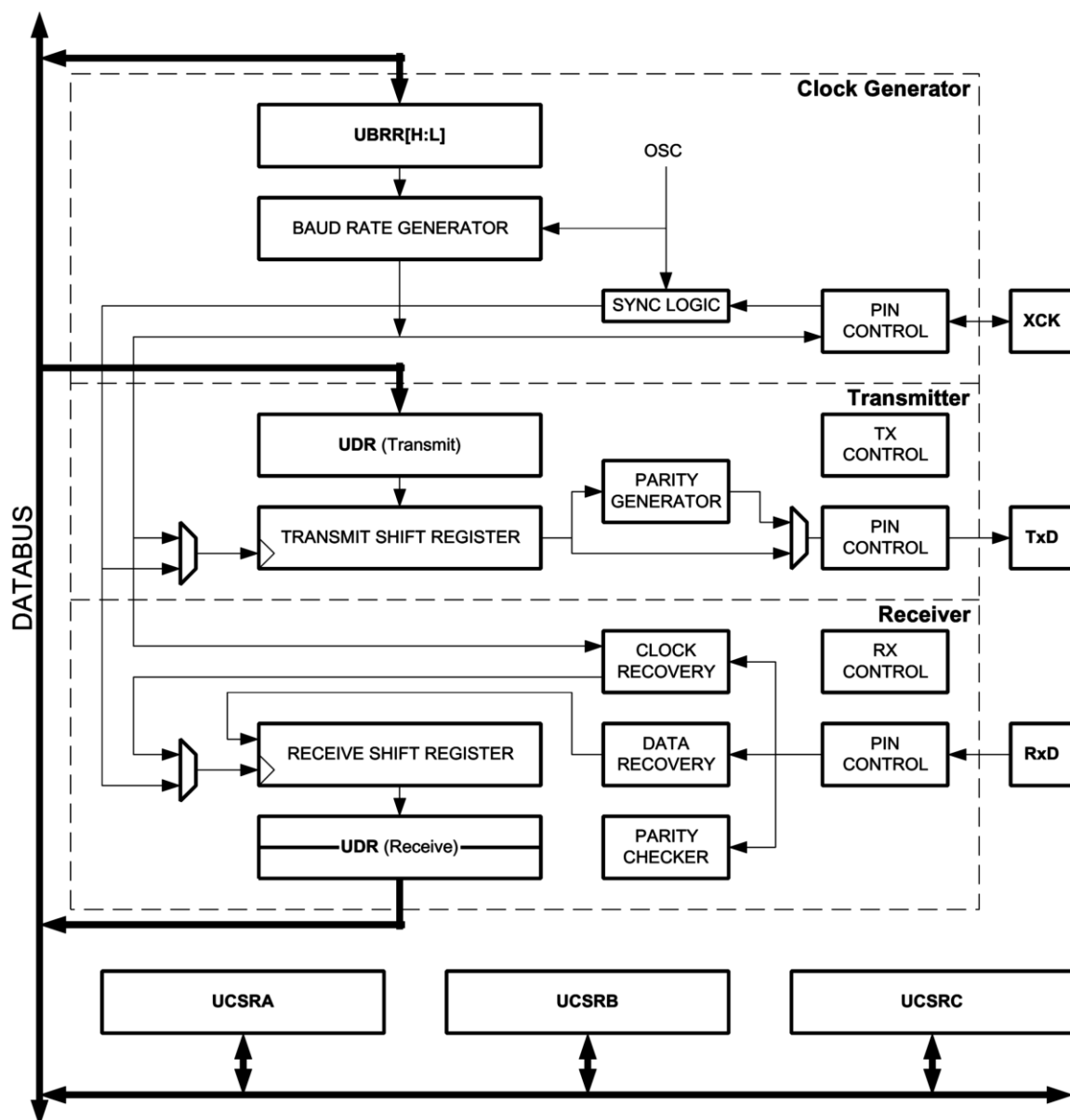


Рис. 176. Блок-схема USART

На схемі (рис. 176) штрих-пунктирною лінією обведені три основні частини USART: синхрогенератор (Clock generator), передавач (Transmitter), приймач (Receiver).

Регістри керування загальні для всіх трьох модулів. Логіка генерації тактового сигналу синхронізації складається з:

- ◇ зовнішнього входу тактового сигналу, що використовуються у веденому режимі;
- ◇ генератора швидкості передачі.

Вхід XCK (тактовий сигнал передачі) використовується тільки в режимі синхронної передачі.

Передавач складається з окремого буфера запису, послідовного зсувного регістра, генератора сигналу парності й логіки контролю для роботи з різними послідовними форматами кадру. Буфер запису забезпечує безперервне передавання даних без затримки між кадрами.

Приймач – найскладніша частина модуля USART через наявність модуля відновлення даних та модуля відновлення тактової частоти. Модулі відновлення використовуються в режимі асинхронного прийому даних. На додаток до модулів відновлення приймач має пристрій перевірки парності, логіку контролю, зсувний регістр і дворівневий буфер прийому (UDR). Приймач підтримує ті ж самі формати кадру, що й передавач, і може виявити помилку кадру, переповнювання даних і помилки парності.

Тактовий генератор виробляє всі основні тактові сигнали — як для передавача, так і для приймача. Модуль USART підтримує чотири режими синхронізації:

- ◇ звичайний асинхронний;
- ◇ асинхронний з подвійною швидкістю;
- ◇ синхронізація від ведучого (Master) пристрою;
- ◇ синхронізація від веденого (Slave) пристрою.

6.11.3.2. Формати кадру

Кадр – це одне слово даних плюс супутні йому біти синхронізації (стартовий біт, стопові біти). Сюди ж може бути доданий біт парності, який застосовується для перевірки правильності передачі інформації.

Канал USART підтримує 30 різних варіантів формату кадру.

Будь-який допустимий формат має такі елементи:

- один стартовий біт;
- 5, 6, 7, 8, або 9 бітів даних;
- біт парності (якщо ввімкнений контроль парності);
- один або два стопових біти.

Кадр починається зі стартового біту, за яким йде молодший розряд слова даних. Потім йде решту інформаційних розрядів. Їх може бути до дев'яти.

Розряди надсилаються у стандартному форматі – від молодшого до старшого.

Якщо перевірка за паритетом увімкнена, то біт парності вставляється між старшим розрядом слова даних і стоповими бітами. Після передавання одного повного кадру канал може відразу починати передавання нового кадру. Якщо новий кадр даних не готовий, канал переходить у режим очікування. Рисунок 177 ілюструє всі можливі комбінації формату кадру. Біти, номери яких розташовані в квадратних дужках, не є обов'язковими.

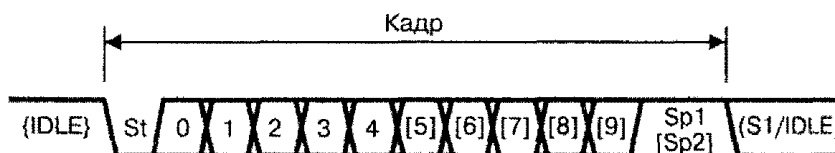


Рис. 177. Формати кадру

Умовні позначення:

St — стартовий біт, завжди рдорівнює нулю;

(n) — біт даних (n = 0—8);

P — біт парності;

Sp — стоповий біт, завжди рдорівнює одиниці;

IDLE— інформація по лінії (RXD або TXD) не передається. В стані IDLE на лінії повинен бути високий логічний рівень.

Таблиця 79

Правила формування біта паритету

Паритет	Кількість одиничних бітів серед даних	Значення біту P
За непарністю	Парне	0
За непарністю	Непарне	1
За парністю	Парне	1
За парністю	Непарне	0

Формат кадру для каналу USART вибирається за допомогою розрядів UCSZ2:0, UPM1:0 і USBS регістрів UCSRB і UCSRC. Для приймача і передавача повинні бути вибрані одні й ті ж установки.

6.11.3.3. Регістри USART

Регістр UDR

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 178. Регістр UDR

Буфер прийому та буфер передачі даних ділять одну і ту ж адресу вводу-виводу і утворюють регістр даних USART, що носить назву UDR (USART Data Register). При записі у регістр UDR дані записуються у відповідний регістр передавача, а при зчитуванні – читаються із регістра приймача.

У випадку приймання 5-, 6- чи 7-бітних символів старші біти регістра заповнюються нулями, а при передаванні відповідні біти ігноруються.

Запис у регістр UDR можливе лише при встановленні біту UDRE у регістрі UCSRA, в іншому випадку дані будуть ігноруватися передавачем UCSRA. Коли дані записуються в буфер передачі і передавач увімкнений, передавач буде завантажувати дані в регістр зсуву, коли регістр зсуву порожній, і біти у відповідному порядку передаються на лінію TxD.

Регістр прийому даних організований складніше і містить дворівневий буфер FIFO, що дозволяє зберігати до двох прийнятих байтів із послідовного порту. Регістр FIFO може змінювати свій стан у процесі виконання програми, тому не варто використовувати при роботі з регістром стану FIFO команди читання-модифікації-запису.

Регістри задавання режиму роботи й стану USART

Для задавання режиму роботи, а також для визначення стану USART використовують три регістри: UCSRA, UCSRB, UCSRC.

Формат регістра UCSRA зображений на рис. 179.

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

Рис. 179. Формат регістра UCSRA

У регістрі UCSRA біти мають таке призначення:

Біт 7 – RXC (Receive Complete). Прийом закінчено. Біт встановлюється в 1, якщо є дані, які були прийняті послідовним портом і ще не зчитувалися. Біт скидається у нуль, якщо буфер приймача порожній і всі дані, що були прийняті, вже зчитані. Біт встановлюється і скидається апаратно і може генерувати переривання "Receive Complete interrupt".

Біт 6 – TXC: (Transmit Complete). Передавання даних завершено. Біт встановлюється в 1, якщо дані передані й немає нових даних для передавання даних в регістрі UDR. Біт TXC автоматично скидається, коли викликається відповідне переривання, або вручну за допомогою запису 0 у відповідну позицію.

Біт 5 – UDRE (USART Data Register Empty). Регістр даних USART порожній. Біт UDRE встановлюється в 1, якщо буфер передавача (UDR) готовий прийняти нові дані. При встановленні біту в одиничний стан може генеруватися відповідне переривання. Після скидання мікросхеми біт встановлюється в одиничне значення, щоб показати, що передавання даних дозволено.

Біт 4 – FE: (Frame Error) Помилка кадру. Біт встановлюється в 1, якщо при прийманні даних виникла помилка формату даних.

Біт 3 – DOR: (Data OverRun). Переповнення регістра даних. Біт встановлюється, якщо виявлено переповнення регістрів даних при прийманні – тобто було прийнято більш ніж два символи і виявлено новий стартовий біт передачі. При записуванні у регістр UCSRA біт завжди потрібно скидати у нуль.

Біт 2 – PE: (Parity Error) Помилка паритету. Встановлюється, якщо у режимі з контролем за парністю/непарністю виявлено помилку паритету. При запису у регістр UCSRA біт завжди потрібно скидати у нуль.

Біт 1 – U2X: (Double the USART Transmission Speed) Встановлення біту в 1 задає подвоєння швидкості передавання даних. Біт використовується лише при асинхронній роботі USART. При синхронному режимі роботи біт має бути у нульовому стані.

Біт 0 – MPCM: (Multi-processor Communication Mode). Режим багатопроцесорних комунікаційних систем. Якщо біт дорівнює 1, вмикається режим багатопроцесорних систем, у котрому всі байти, які не містять відповідними чином встановленого адресного біту, ігноруються.

Формат регістра UCSRB показано на рис. 180.

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 180. Формат регістра UCSRB

У регістрі UCSRB біти мають таке призначення:

Біт 7 – RXCIE: (RXC Interrupt Enable). Дозвіл переривання за закінченням приймання даних. Запис 1 у біт дозволяє переривання від біту RXC регістра UCSRA.

Біт 6 – TXCIE: (TXC Interrupt Enable). Дозвіл переривання за завершенням передавання даних. Встановлення біту в 1 дозволяє переривання від біту TXC з регістра UCSRA.

Біт 5 – UDRIE: (USART Data Register Empty Interrupt Enable). Дозвіл переривання при порожньому регістрі даних USART. Встановлення біту в одиничний стан дозволяє переривання від біту UDRE.

Біт 4 – RXEN: (Receiver Enable). Дозвіл роботи приймача. Встановлення біту в 1 дозволяє роботу приймача даних USART та перемикає роботу відповідної лінії мікросхеми на режим роботи з USART. Встановлення біту в 0 очищає буфер приймача, переводить у недійсний стан біти FE, DOR та PE, а також перемикає лінію RxD мікросхеми на нормальний режим роботи.

Біт 3 – TXEN: (Transmitter Enable). Дозвіл роботи передавача. Встановлення біту в 1 дозволяє роботу передавача USART та переводить лінію

TxD у режим роботи з USART. Скидання біту TXEN вимикає передавач лише після передавання байтів, що мають бути передані.

Біт 2 – UCSZ2: (Character Size). Біт довжини слова, що передається. Біт використовується разом із бітами UCSZ1:0 з регістра UCSRC для задавання довжини слова, що передається.

Біт 1 – RXB8: (Receive Data Bit 8). Задає дев'ятий біт даних, що приймаються. Для вірного приймання даних біт RXB8 має бути прочитаний до читання даних з регістра UDR.

Біт 0 – TXB8: (Transmit Data Bit 8). Задає дев'ятий біт даних, що передається при роботі із 9-бітовими кодами. Для вірної роботи біт має бути записаний у регістр UDR до запису даних.

Формат регістра UCSRC показано на рис. 181.

Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	

Рис. 181. Формат регістра UCSRC

Регістр UCSRC має спільну адресу з регістром UBRRH. Тип регістра при запису визначається бітом URSEL, а при читанні визначається послідовністю читань, яка буде розглянута нижче.

У регістрі UCSRC біти мають таке призначення:

Біт 7 – URSEL: (Register Select). Біт вибору регістра. Регістри UCSRC та UBRRH суміщають єдину адресу. При записі даних у регістр UCSRC біт має дорівнювати 1. При читанні цей біт дорівнює 1, а решта бітів відповідають регістру UCSRC.

Біт 6 – UMSEL: (USART Mode Select). Вибір режиму роботи послідовного порту. Одиничне значення біту переводить послідовний порт у синхронний режим роботи, а нульове значення – в асинхронний режим.

Біти 5 та 4 – UPM1:0: (USART Parity Mode). Біти вибору режиму контролю за парністю чи непарністю. Режим контролю задається згідно з таблицею 80.

Таблиця 80

Режим контролю за паритетом

UPM1	UPM0	Тип контролю
0	0	Від'єднаний. Біту контролю за паритетом немає
0	1	Зарезервовано для подальших модифікацій
1	0	Увімкнений контроль за парністю. Біт паритету дорівнює 1, якщо серед інформаційних бітів парна кількість одиниць
1	1	Увімкнений контроль за непарністю. Біт паритету дорівнює 1, якщо серед інформаційних бітів непарна кількість одиниць

При увімкненому контролі за паритетом при передаванні даних автоматично формується відповідний біт паритету, а при прийманні даних контролюється стан відповідного біту. При невідповідності біту паритету у прийнятих даних із розрахунковим для даного режиму встановлюється біт PE у регістрі UCSRA.

Біт 3 – USBС: (USART Stop Bit Select). Вибір кількості стоп-бітів. Кількість стоп-бітів визначається за таблицею 81.

Таблиця 81

Кількість стоп-бітів залежно від стану USBС

USBС	Кількість стоп-бітів
0	1 стоп-біт
1	2 стоп-біти

Таблиця 82

Кількість інформаційних бітів залежно від стану бітів UCSZ2-0

UCSZ2	UCSZ1	UCSZ0	Кількість інформаційних бітів, що передаються чи приймаються
0	0	0	5 бітів
0	0	1	6 бітів
0	1	0	7 бітів
0	1	1	8 бітів
1	0	0	Зарезервовано
1	0	1	Зарезервовано
1	1	0	Зарезервовано
1	1	1	9 бітів

Біти 2 та 1 – UCSZ1:0 (USART Character Size). Біти розміру символу.

Разом із бітом UCSZ2 із регістра UCSRB визначають кількість інформаційних бітів, що передаються та приймаються через USART.

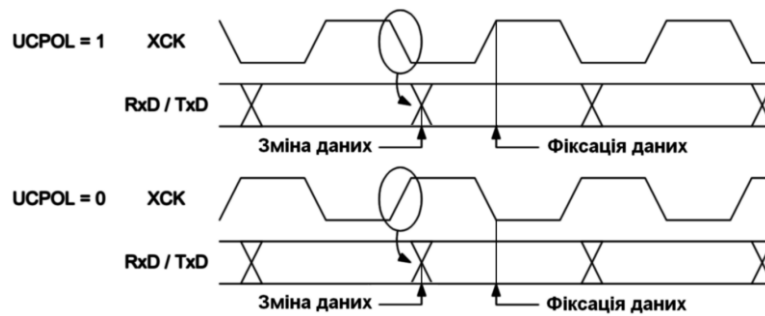


Рис. 182. Відповідність тактового сигналу і даних залежно від стану біту UCPOL

Біт 0 – UCPOL: (USART Clock Polarity). Полярність тактового сигналу. Біт має значення лише у синхронному режимі роботи. В асинхронному режимі роботи біт не має значення і, зазвичай, встановлюється в нульовий стан. Полярність тактового сигналу визначається згідно з рис. 182.

Регістри задавання швидкості приймання й передавання UBRRL та UBRRH

Регістри UBRRL та UBRRH призначені для задавання швидкості передавання та приймання даних.

Формат регістрів зображений на рис. 183.

Bit	15	14	13	12	11	10	9	8	
	URSEL	-	-	-	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Рис. 183. Формат регістрів UBRRL та UBRRH

Регістр UBRRH має однакову адресу із регістром UCSRC, а регістр, до якого відбувається звертання, визначається бітом URSEL.

Біт 15 – URSEL (USART Register Select) – біт визначення регістра. Одиничне значення біту вказує, що вибрано регістр UCSRC, нульове – на вибір регістра UBRRH.

Біти 14:12 – зарезервовані для подальшого використання. Для сумісності у ці біти мають бути записані нулі.

Біти 11:0 – UBRR11:0: (USART Baud Rate Register). Регістр вибору швидкості передавання та приймання даних через USART.

Залежність швидкості прийому та передачі наведена у таблиці 83.

Таблиця 83

Швидкість прийому та передачі

Режим роботи	Значення регістра залежно від швидкості	Швидкість
Асинхронний з нормальною швидкістю	$UBRR = \frac{f_{osc}}{16 \text{ BAUD}} - 1$	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$
Асинхронний з подвоєною швидкістю	$UBRR = \frac{f_{osc}}{8 \text{ BAUD}} - 1$	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$
Синхронний режим	$UBRR = \frac{f_{osc}}{2 \text{ BAUD}} - 1$	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$

Запис у регістр проводять у такій послідовності: спочатку записують старші біти у регістр UBRRH, а потім – дані у регістр UBRRL. Зміна коефіцієнта UBRR11:0 здійснюється у момент запису даних у регістр UBRRL.

Доступ до регістрів UBRRH UCSRC

Регістри UBRRH та UCSRC мають однакову адресу. Для доступу при запису даних використовують старший біт регістрів UBRRH та UCSRC URSEL. Якщо біт дорівнює нулю, решта бітів регістра записується у регістр UBRRH, якщо ж біт URSEL дорівнює одиниці, змінюється значення регістра UCSRC. Наступна підпрограма ілюструє запис у два регістри:

```
; встановити у UBRRH код 2, а у регістр UCSRC код 0ff
ldi R16,0x02;
ldi R17,0x0ff
out UBRRH, R16; завантажуюмо значення у регістр UBRRH
out UBRRH, R17; завантажуюмо значення у регістр UCSRC
```

Читання регістрів є складнішою операцією, хоча більшості програм читання регістрів UBRRH та UCSRC не потрібне. Доступ на читання контролюється послідовністю читань.

Якщо йде підряд дві команди читання регістра з адресою UCSRC/UBRRH, то перший раз буде прочитане значення регістра UBRRH, а другий раз – регістра UCSRC. У випадку, коли між командами читання буде хоча б одна команда, відбудеться читання два рази підряд лише регістра UBRRH.

Переривання у процесі читання регістрів UCSRC/UBRRH мають бути заблоковані.

Приклад читання регістрів UCSRC/UBRRH (переривання заборонені)

```
USART_ReadUCSRC:  
IN R16, UBRRH ; читаємо регістр UBRRH  
IN R17, UCSRC ; і вдіразу читаємо регістр UCSRC  
ret
```

6.11.3.4. Ініціалізація послідовного порту

USART має бути ініціалізована до будь-якого передавання чи приймання даних. Процес ініціалізації складається зі встановлення швидкості передавання, встановлення формату кадру і дозволу передавача чи приймача (або обох одразу). Для дозволу переривань глобальний дозвіл переривань на час ініціалізації повинен бути скинутий.

Перед повторною ініціалізацією чи зміною режиму порту слід переконатися, що передавання даних завершена, а у буфері приймання не залишилося потрібних даних. Простий код ініціалізації наведено нижче:

```
USART_Init:  
; Регістри R17, R16 задають швидкість передавання  
даних  
; Регістр R18 задає стан UCSRB, а  
; Регістр R19 - стан UCSRC  
; задаємо швидкість передавання даних  
OUT UBRRH, R17  
OUT UBRRL, R16  
; Задаємо формат кадра  
OUT UCSRC, R19  
; Дозволяємо приймач і передавач  
ORI R18, (1<<RXEN) | (1<<TXEN)  
OUT UCSRB, R18  
RET
```

6.11.3.5. Процедури передавання та приймання даних

Передавання даних через послідовний порт

Передавання даних ініціюється завантаженням у буфер передавання UDR даних, однак для роботи виводу TXD у режимі передавача необхідно

дозволити передавач встановленням біту TXEN (Transmit Enable). Якщо попереднє передавання даних завершено, то дані відразу надійдуть у зсувний регістр, якщо попереднє передавання не завершилося – дані із регістра UDR будуть переміщені в зсувний регістр, коли регістр зсуву буде порожнім.

Наступний приклад показує простий вивід даних у послідовний порт USART із опитуванням біту UDRE. При використанні форматів даних менше, ніж з 8 бітами даних, старші біти, записані в UDR, ігноруються. Послідовний порт має бути проініціалізований до передавання даних. Наведена функція очікує, коли буфер передачі стане порожнім за допомогою перевірки біту UDRE перед завантаженням нових даних. Дані для передавання записані у регістрі R16.

```
USART_Transmit:
; Очікування звільнення буфера передавача
USART_Transmit_Ready:
sbis UCSRA,UDRE      ; пропусти наступну команду,
                    ; якщо буфер порожній
rjmp USART_Transmit_Ready
                    ; перейти на очікування
                    ; звільнення передавача
out UDR,r16          ; помістити дані у буфер
                    ; передавача
                    ; та почати пересилання даних
ret
```

Зрозуміло, що підпрограма передавання даних може бути переписана так, щоб, за відсутності можливості передавати дані, вихід з підпрограми відбувався миттєво із встановленням біту перенесення.

Прийом даних через USART

Приймач USART вмикається записом у біт (RXEN) (Receive Enable) регістра UCSRB одиниці. Коли приймач увімкнений, нормальна робота послідовного інтерфейсу припиняється і лінія переходить у режим роботи з послідовним портом. Швидкість передавання даних, режим роботи і формат кадру встановлюється однаковими для передавача і приймача. У синхронному режимі роботи при використанні синхронного керування вхід ХСК використовується для передавання чи приймання тактового сигналу.

Наступний приклад показує найпростіший варіант приймання даних, що не потребує переривання, а здійснює опитування біту RXC (Receive Complete). На відміну від підпрограми передавання, у підпрограмі приймання даних не здійснюється очікування наявності даних. Якщо даних немає, то встановлюється одиничне значення біту перенесення. У випадку наявності даних дані приймаються у регістр R16, а біт перенесення скидається. Перед використанням функції послідовний порт має бути дозволений і проініціалізований.

```

USART_Receive:
SBIS UCSRA, RXC ; Аналіз біту RXC,
                  ; Якщо дані є, пропустити наступну
                  ; команду
RJMP USART_Not_Receive
IN R16, UDR ; Отримати дані
CLC ; Скинути ознаку перенесення
RET ; Повернутися у основну програму
USART_Not_Receive: ; Даних немає
SEC ; Встановити біт перенесення
RET ; Повернутися у основну програму

```

Зрозуміло, що підпрограма приймання даних може бути переписана так, щоб за відсутності даних відбувалося очікування надходження даних.

6.11.3.6. Багатопроцесорні системи і передавання 9 бітів

У випадку взаємодії багатопроцесорних систем можливе використання режиму 9-бітового передавання з апаратною фільтрацією кадрів.

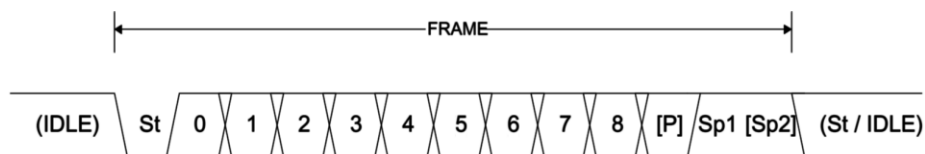


Рис. 184. Формат кадру 9-бітової передачі

Для такого використання останній біт даних (восьмий) встановлюється в 1 при передаванні адреси та дорівнює нулю при передаванні даних. Ведені контролери, що приймають кадри у такому режимі, ігнорують усі кадри зі скинутим останнім бітом даних. При отриманні кадру із адресою, що співпадає із заданою веденому контролеру, контролер переходить у режим прийому всіх байтів і приймає чи передає задані дані, а потім повертається у режим з апаратною фільтрацією кадрів.

6.12. АЦП мікроконтролера

6.12.1. Загальна характеристика АЦП мікросхеми

Мікросхема АТМega32 у своїй структурі має АЦП із такими властивостями:

- ◇ Роздільна здатність 10 біт.
- ◇ Інтегральна нелінійність дорівнює 0,5 ваги молодшого розряду.
- ◇ Абсолютна точність ± 2 .
- ◇ Час перетворення 13 – 260 мкс.
- ◇ Частота перетворення до 15 кГц при максимальній роздільній здатності.
- ◇ Кількість несиметричних входів 8.
- ◇ Кількість диференціальних входів 7.
- ◇ Кількість диференціальних каналів з вибором коефіцієнта підсилення 2.
- ◇ Діапазон вхідних напруг для перетворення 0-VCC.
- ◇ Опорна напруга зовнішня і внутрішня.
- ◇ Режими перетворення: циклічне перетворення, із запуском за сигналом чи подією, програмний запуск.

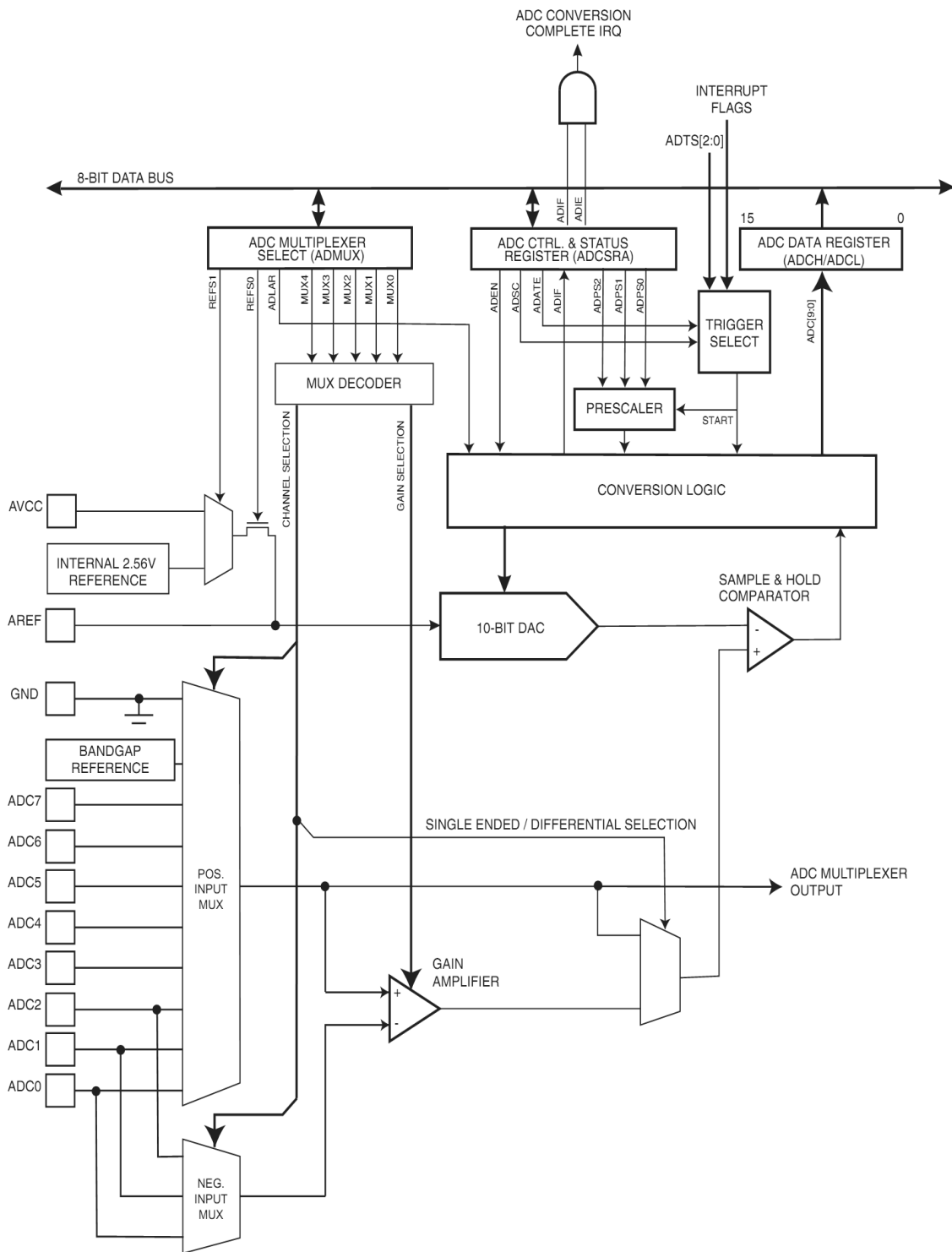


Рис. 185. Структурна схема АЦП

Блок-схема АЦП АТМega32 наведена на рис. 185. Основою блоку АЦП є одноканальний 10-розрядний АЦП послідовного наближення, що побудований за класичною схемою на основі 10-бітного ЦАП (10-bit DAC), компаратора із запам'ятовуванням стану (Sample & hold comparator) та відповідного регістра послідовного наближення з логікою (Conversion logic). Синхронізація

перетворення здійснюється за допомогою тактової частоти процесора, що може ділитися попереднім подільником (Prescaler). Запуск та зупинка перетворення здійснюються за допомогою тригерної схеми (Trigger select).

Необхідна для перетворювача опорна напруга може задаватися або внутрішнім джерелом опорної напруги Vref напругою 2,56 В (Internal 2,56V reference), або входом AREF чи напругою живлення аналогової частини схеми AVCC. Вибір потрібного джерела здійснюється сигналами REFS1 та REFS0. Вхід AREF може використовуватися як вихід для контролю рівня опорної напруги або для подальшої її фільтрації.

Вхідний сигнал для АЦП може бути як однополярним, так і диференціальним. Джерела вхідного сигналу, що подається на контакти порту А, обираються мультиплексорами для позитивного (Pos. input mux) і негативного сигналів (Neg. input mux). АЦП підтримує режим 10 однополярних та 16 диференціальних вхідних комбінацій напруги. Серед джерел вхідного сигналу для позитивного входу можуть бути також опорна напруга забороненої зони напівпровідника (Bandgap reference) та напруга на земляному вході, що можуть бути використані для калібрування АЦП. Вибір джерела та типу сигналу здійснюється дешифратором мультиплексорів (Mux decoder). АЦП володіє можливістю внутрішнього підсилення диференціальних сигналів за допомогою підсилювача (Gain amplifier) із фіксованим коефіцієнтом підсилення, що забезпечує підсилення на 20 дБ, чи 46 дБ (тобто x10 або x200). Для контролю роботи, вибору режиму та читання результатів перетворення використовуються регістри:

- ◇ ADC multiplexer select (ADMUX) – регістр вибору вхідного мультиплексора;
- ◇ ADC ctrl. & status register (ADCSRA) – регістр контролю та статусу АЦП;
- ◇ ADC data register (ADCH/ADCL) – регістр даних АЦП;
- ◇ SFIOR Special FunctionIO Register – регістр спеціальних функцій вводу-виводу, у котрому використовуються біти ADTS2:0.

Мінімальне значення коду на виході АЦП відповідає напрузі, яка дорівнює 0 В, а значення на 1 більше, ніж максимально можливе, представляє напругу на вході, що використовується як опорна.

6.12.2. Робота АЦП

АЦП вмикається встановленням біту дозволу АЦП ADEN у регістрі ADCSRA. Без встановлення сигналу ADEN опорна напруга та вхідний сигнал АЦП не будуть набувати вірних значень. Таку особливість варто мати на увазі при роботі із зовнішнім фільтруючим конденсатором на вході AREF. Коли біт ADEN скинутий, АЦП повністю вимкнений і не споживає енергії. При переході у режим енергозбереження рекомендується вимкнути АЦП.

Одиничне перетворення запускається шляхом запису логічної одиниці в біт запуску перетворення ADSC (ADC Start Conversion) регістра ADCSRA. Цей біт залишається високим, поки перетворення триває, і апаратно очищається при завершенні перетворення. Якщо вибрано інший канал передавання даних у той час, коли триває перетворення, АЦП закінчить поточне перетворення перед виконанням зміни каналу.

Проте перетворення може запускатися автоматично при різних подіях. Автоматичне ввімкнення вмикається установкою біту ADATE (ADC Auto Trigger Enable) у регістрі ADCSRA. Джерело запуску обирається шляхом встановлення бітів ADTS у регістрі SFIOR.

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 186. Формат регістра SFIOR

Запуск здійснюється позитивним фронтом відповідного сигналу. Джерела запуску АЦП визначаються за таблицею 84.

Джерела сигналу запуску перетворення АЦП

ADTS2:0	Джерела запуску АЦП
0 0 0	Вільний запуск. Наступне перетворення запускається за закінченням поточного (Free Running mode)
0 0 1	Аналоговий компаратор (Analog Comparator)
0 1 0	Зовнішній вхід переривань 0 (External Interrupt Request 0)
0 1 1	Таймер-лічильник 0 переривання за рівністю (Timer/Counter0 Compare Match)
1 0 0	Таймер-лічильник 0 переривання за переповненням (Timer/Counter0 Overflow)
1 0 1	Таймер-лічильник 1 переривання за рівністю В (Timer/Counter1 Compare Match B)
1 1 0	Таймер-лічильник 1 переривання за переповненням (Timer/Counter1 Overflow)
1 1 1	Таймер-лічильник 1 переривання за захопленням (Timer/Counter1 Capture Event)

Схема сигналів запуску АЦП наведена на рис. 187.

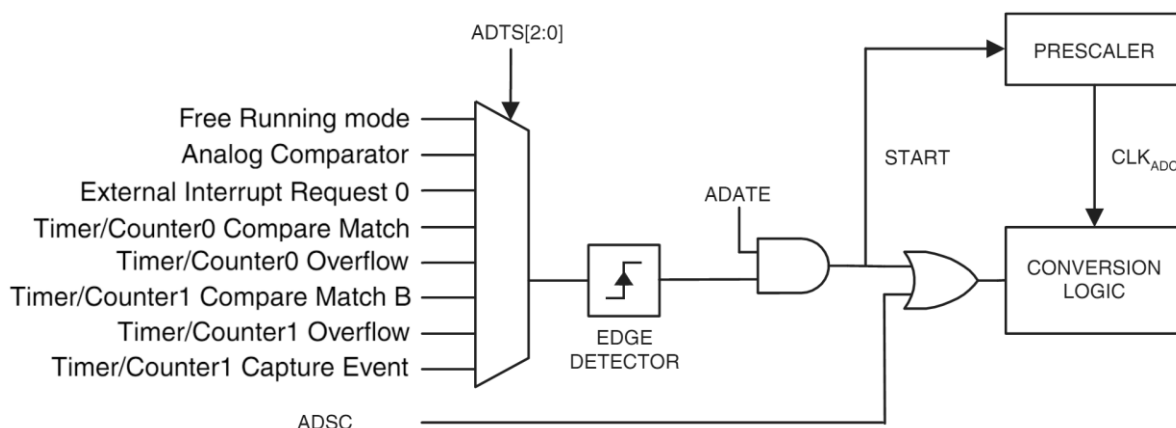


Рис. 187. Логіка формування сигналу запуску АЦП

Зауважимо, що при запуску АЦП скидається попередній подільник АЦП, який забезпечує фіксований час перетворення. Якщо сигнал запуску залишається встановленим після завершення перетворення, нове перетворення не буде запущено. Також ігнорується повторний позитивний фронт, якщо перетворення не завершилося.

За закінченням перетворення АЦП, незалежно від того, яким методом було воно запущено, встановлюється ознака переривання за закінченням перетворення АЦП. Ознака запиту переривання від АЦП буде встановлена, навіть якщо певне переривання заборонене або скинутий біт глобального

дозволу переривання у регістрі SREG. При забороні переривання перетворення може бути викликане таким чином, не викликаючи переривання. Проте, якщо ознака переривання буде використовуватися у подальшому для контролю за станом АЦП, вона має бути скинута у нульовий стан програмним шляхом.

6.12.3. Попередній подільник та частота перетворення АЦП

АЦП у даній мікросхемі є досить повільним вузлом. За замовчуванням схема послідовного наближення вимагає тактову частоту між 50 кГц і 200 кГц для отримання максимальної роздільної здатності. Таке обмеження пов'язане із низькою швидкістю аналогового компаратора та блоку ЦАП.

Якщо немає потреби у 10-бітній роздільній здатності, вхідна тактова частота АЦП може бути вищою за 200 кГц для отримання вищої частоти дискретизації, проте у багатьох випадках при роботі на максимальній частоті потрібна висока точність перетворення. Для узгодження частоти роботи ядра мікроконтролера і модуля АЦП у модуль АЦП введено попередній подільник частоти, який знижує частоту роботи модуля і вводить її у потрібний діапазон. Коефіцієнт поділу попереднього подільника програмується і визначається бітами ADPS у регістрі ADCSRA.

Попередній подільник починає відлік з моменту вмикання АЦП за допомогою встановлення біту ADEN у регістрі ADCSRA. Попередній подільник продовжує працювати до тих пір, поки біт ADEN встановлений, і скидається при скиданні біту ADEN, а також при кожному старті перетворення.

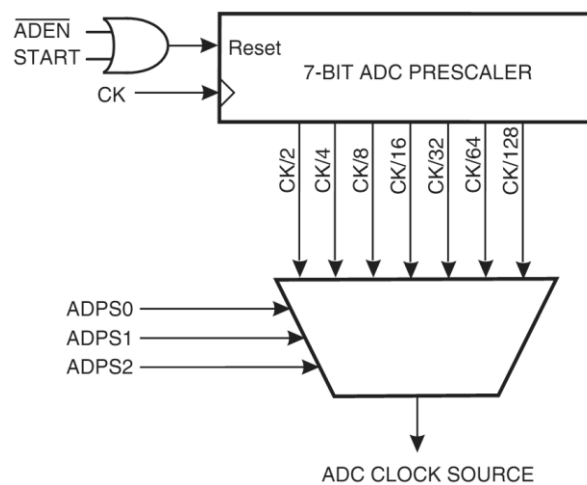


Рис. 188. Попередній подільник АЦП

Така особливість дозволяє витримати фіксовану тривалість перетворення. Формат регістра ADCSRA зображено на рис. 189.

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 189. Регістр контролю та статусу АЦП ADCSRA

Коефіцієнт поділу попереднього подільника залежно від бітів ADPS визначається за таблицею 85. При ініціалізації перетворення встановленням біту ADSC у регістрі ADCSRA, перетворення починається за наступним зростаючим фронтом тактового сигналу АЦП. На початку перетворення АЦП виділяє певний час для захоплення рівня аналогового сигналу. Тривалість перетворення залежить від режиму роботи АЦП. Сумарний час перетворення для різних режимів роботи у періодах тактового сигналу наведено у таблиці 86.

Таблиця 85

Коефіцієнт поділу попереднього подільника АЦП

ADPS 2:0	Коефіцієнт поділу попереднього подільника АЦП
0 0 0	2
0 0 1	2
0 1 0	4
0 1 1	8
1 0 0	16
1 0 1	32
1 1 0	64
1 1 1	128

Таблиця 86

Час перетворення для різних режимів роботи

Режим роботи	Час захоплення сигналу	Сумарний час перетворення
Перше однократне перетворення в новому каналі, однополярний сигнал	13.5	25
Однократне перетворення, однополярний сигнал	1.5	13
Циклічне перетворення, однополярний сигнал	2	13.5
Однократне перетворення, диференціальний сигнал	1.5/2.5	13/14
Циклічне перетворення, диференціальний сигнал	2.5	14
Перше однократне перетворення в новому каналі, диференціальний сигнал	13.5	25

Нормальне перетворення займає 13 тактів. Перше перетворення після ввімкнення АЦП (АДЕН в ADCSRA встановлений) займає 25 ADC тактів для ініціалізації аналогових схем. Мінімальна тривалість вибірки сигналу триває 1,5 такта АЦП. У режимі однократного перетворення диференціального сигналу тривалість вибірки АЦП може становити або 1,5, або 2,5 такта тактового сигналу АЦП, і залежить від моменту надходження сигналу запуску АЦП. Тому тривалість однократного перетворення диференціального сигналу становить або 13, або 14 тактів. При переході на інший канал перша вибірка АЦП триває 13,5 такта, тому повна тривалість перетворення при переході із каналу в канал триває 25 тактів роботи АЦП.

При виборі тактової частоти АЦП слід пам'яті, що схема внутрішнього підсилювача оптимізована для ширини смуги в 4 кГц на всіх рівнях підсилення. При передаванні вищих частот можуть виникнути спотворення. Для їх уникнення необхідно використовувати зовнішній фільтр нижніх частот. Слід зазначити, що можлива тактова частота АЦП не залежить від коефіцієнта підсилення внутрішнього підсилювача.

6.12.4. Джерела сигналів АЦП

Зміна активного каналу (чи каналів) АЦП здійснюється бітами MUXn та REFS1:0 у регістрі ADMUX. Формат регістра ADMUX, що задає джерела опорної напруги та вхідного сигналу або сигналів, показано на рис. 190. Призначення бітів, що відповідають вибору джерел сигналу, наведено у наступній таблиці.

7	6	5	4	3	2	1	0	
REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

Рис. 190. Формат регістра ADMUX

Таблиця 87

Джерела напруги, яка перетворюється		
MUX4.0	Джерело напруги, яка перетворюється	Коефіцієнт підсилення
0 0000	Сигнал ADC0, недиференціальний вхід	1
0 0001	Сигнал ADC1, недиференціальний вхід	1
0 0010	Сигнал ADC2, недиференціальний вхід	1
0 0011	Сигнал ADC3, недиференціальний вхід	1
0 0100	Сигнал ADC4, недиференціальний вхід	1
0 0100	Сигнал ADC5, недиференціальний вхід	1
0 0110	Сигнал ADC6, недиференціальний вхід	1
0 0111	Сигнал ADC7, недиференціальний вхід	1
0 1000	Різниця сигналів ADC0- ADC0, диференціальний вхід 1)	10
0 1001	Різниця сигналів ADC1- ADC0, диференціальний вхід	10
0 1010	Різниця сигналів ADC0- ADC0, диференціальний вхід 1)	200
0 1011	Різниця сигналів ADC1- ADC0, диференціальний вхід	200
0 1100	Різниця сигналів ADC2- ADC2, диференціальний вхід 1)	10
0 1101	Різниця сигналів ADC3- ADC2, диференціальний вхід	10
0 1110	Різниця сигналів ADC2- ADC2, диференціальний вхід 1)	200
0 1111	Різниця сигналів ADC3- ADC2, диференціальний вхід	200
1 0000	Різниця сигналів ADC0- ADC0, диференціальний вхід 1)	1
1 0001	Різниця сигналів ADC1- ADC0, диференціальний вхід	1
1 0010	Різниця сигналів ADC2- ADC0, диференціальний вхід	1
1 0011	Різниця сигналів ADC3- ADC0, диференціальний вхід	1
1 0100	Різниця сигналів ADC4- ADC2, диференціальний вхід	1
1 0101	Різниця сигналів ADC5- ADC2, диференціальний вхід	1
1 0110	Різниця сигналів ADC6- ADC2, диференціальний вхід	1
1 0111	Різниця сигналів ADC7- ADC2, диференціальний вхід	1
1 1000	Різниця сигналів ADC0- ADC2, диференціальний вхід	1
1 1001	Різниця сигналів ADC1- ADC2, диференціальний вхід	1
1 1010	Різниця сигналів ADC2- ADC2, диференціальний вхід 1)	1
1 1011	Різниця сигналів ADC3- ADC2, диференціальний вхід	1
1 1100	Різниця сигналів ADC4- ADC2, диференціальний вхід	1
1 1101	Різниця сигналів ADC5- ADC2, диференціальний вхід	1
1 1110	1,22 В (Ширина забороненої зони напівпровідника)	1
1 1111	0 В	1
	1) Використовується для калібрування встановлення 0 2) Використовується для визначення параметрів температури	

Як бачимо з таблиці, будь-які входи АЦП, а також сигнал опорної напруги, можуть бути обрані в якості несиметричних входів АЦП.

Регістр ADMUX має тимчасовий буферний регістр, що запам'ятовує дані на час перетворення. Це гарантує, що зміна каналу й коефіцієнта підсилення, який використовується, відбувається лише в безпечний момент. Після початку перетворення зміна каналу й коефіцієнта підсилення блокується для забезпечення достатнього часу вибірки АЦП. Отже, зміна стану регістра ADMUX у процесі перетворення не впливає на поточний цикл перетворення і вступає в силу лише після його завершення. Проте не варто відразу змінювати стан регістра ADMUX після встановлення біту запуску АЦП (біту ADSC), так як перетворення починається лише наступного тактового імпульсу АЦП після встановлення біту ADSC, і тактовий імпульс може надійти пізніше, ніж буде записане нове значення у регістрі ADMUX.

У випадку використання запуску АЦП за сигналом, точний час ініціюючої події може бути неоднозначним. Особлива увага повинна бути прийнята при оновленні регістра ADMUX, щоб визначити, яке саме перетворення буде викликане новими налаштуваннями. Також варто звернути увагу на можливість переривання у процесі запису в регістр ADMUX. Якщо обидва біти ADATE і ADEN дорівнюють 1, перетворення може початися у будь-який час. Якщо регістр ADMUX змінюється в цей період, користувач не може сказати, з якого каналу відбувається перетворення сигналу.

Тому часто при роботі у режимі запуску АЦП за сигналом варто зупинити АЦП, змінити налаштування і потім запустити АЦП знову. Якщо це неможливо, ADMUX можна оновлювати, якщо:

1. ADATE або ADEN скинуті.
2. Під час перетворення мінімум через один такт АЦП після події запуску.
3. Після перетворення, перш ніж ознака переривання буде використана як джерело запуску.

При оновленні ADMUX поза однієї із цих умов, нові налаштування будуть впливати на наступний цикл перетворення АЦП.

Після того, як диференціальний канал обраний, перехідний процес може тривати цілих 125 мкс до стабілізації на новому значенні. Таким чином, перетворення не повинно бути розпочате протягом перших 125 мкс після вибору нового диференціального каналу. Крім того, результати перетворення, отримані протягом цього періоду, повинні бути відкинуті.

При зміні вибору каналу користувач повинен дотримуватися таких правил, аби гарантувати, що обрано правильний канал: у режимі одиничного перетворення завжди спочатку варто обирати канал, перш ніж почати перетворення. Вибір каналу може бути здійснено через один тактовий цикл АЦП після встановлення біту ADSC. Проте найпростіший спосіб – це чекати закінчення перетворення, перш ніж змінювати вибір каналів.

В автономному режимі завжди потрібно вибрати канал, перш ніж почати перше перетворення. Вибір каналу може бути здійснено через один тактовий цикл АЦП після запису одиниці в ADSC. Але найкраще дочекатися закінчення першого перетворення, а потім змінити вибір каналів. Оскільки наступне перетворення вже почалося автоматично, наступний результат буде відображати попередній вибір каналу. Наступне перетворення буде відображати новий вибір каналу.

При перемиканні диференціального каналу із підсиленням, перший результат перетворення може мати низьку точність. Користувач повинен ігнорувати перший результат перетворення.

6.12.5. Вибір опорної напруги

Таблиця 88

Джерела опорної напруги

REFS1	REFS0	Джерело опорної напруги
0	0	Зовнішній сигнал із лінії AREF у діапазоні 2,0 В - AVCC, внутрішнє джерело напруги Vref вимкнене
0	1	AVCC з зовнішнім конденсатором на вході AREF у якості фільтра
1	0	Зарезервовано. Фактично внутрішнє джерело напруги Vref ввімкнене, але його напруга вимкнена від виводу AREF, на котрий можна подавати зовнішній сигнал опорної напруги. З точки зору прикладного використання, цей режим нічим не відрізняється від режиму REFS1:0 = 00
1	1	Внутрішнє джерело напруги Vref з напругою 2,56 В із можливістю використання зовнішнього конденсатора на вході AREF у якості фільтра

Для роботи АЦП необхідна опорна напруга, з котрою здійснюється порівняння напруги, що перетворюється. АЦП ATmega32 може працювати як із кількома внутрішніми джерелами опорної напруги, так і з зовнішнім сигналом опорної напруги. Вибір джерела опорної напруги здійснюється бітами REFS1:0 у регістрі ADMUX. Формат регістра ADMUX, що задає джерела опорної напруги, наведено на рис. 190, а призначення бітів, що відповідають вибору джерела опорної напруги, – у таблиці 88.

Сигнал AVCC та внутрішня опорна напруга Vref під'єднуються до АЦП та виводу AREF через пасивний комутатор без будь-якого підсилення за потужністю. Максимальний струм лінії AREF становить 100 мкА при напрузі живлення 2,7 В і 175 мкА – при напрузі живлення 5 В.

Так як джерело внутрішньої опорної напруги Vref має високий імпеданс, то до лінії AREF у цьому режимі роботи можна під'єднувати виключно високоомні входи або фільтруючі конденсатори. І хоча сигнал AVCC має велику потужність, все одно не слід використовувати вивід AREF для використання із низькими приймачами сигналів.

Також слід бути уважним при під'єднанні зовнішнього джерела опорної напруги до виводу AREF. При використанні зовнішнього джерела опорної напруги не слід використовувати інші варіанти ввімкнення опорної напруги, так як вони будуть замкнуті на зовнішню напругу.

6.12.6. Зниження шумів перетворення

У мікросхемі ATМega32 АЦП має можливість перетворення у сплячому режимі для зменшення шумів, індукованих ядром процесора та іншими периферійними пристроями вводу/виводу. Для використання перетворення у сплячому режимі потрібно скористатися такою процедурою:

1. Необхідно перевести АЦП у режим однократного перетворення та глобально дозволити переривання і переривання від АЦП за закінченням перетворення. Для уникнення проблем із шумом бажано встановити виключно глобальний дозвіл переривання і дозвіл переривання від АЦП.

2. Перевести контролер у режим ADC Noise Reduction (або в режим очікування). АЦП почне перетворення, як тільки процесор буде зупинений.

3. Якщо інші переривання не надійдуть до завершення перетворення АЦП, переривання від АЦП виведе процесор із режиму очікування. Якщо ж інші переривання не заблокувати, то процесор прокинеться до закінчення перетворення в АЦП і на результат перетворення накладеться шум від роботи блоків мікросхеми.

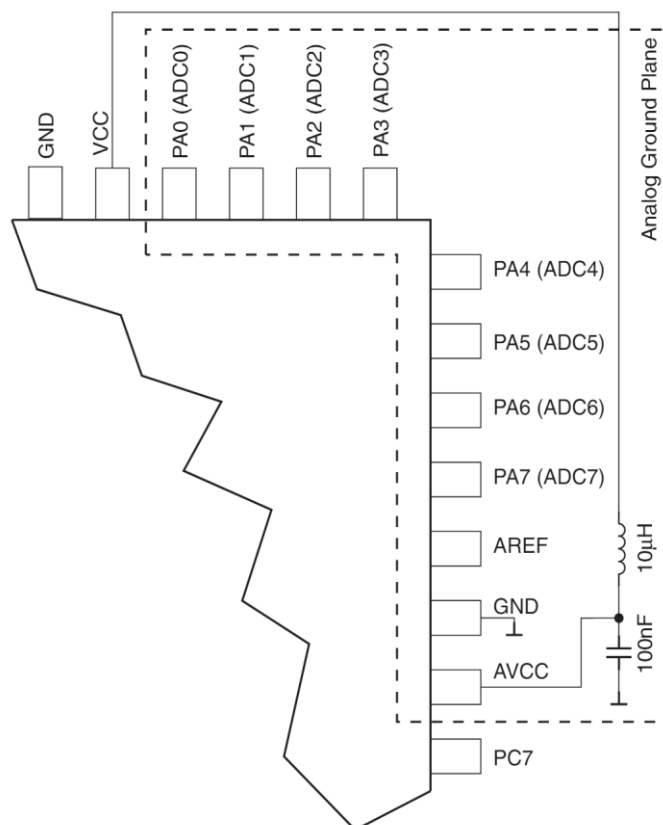


Рис. 191. Під'єднання живлення аналогової частини мікросхеми

Також варто звернути увагу на те, що мікросхема і зовнішні схеми за межами корпусу мікросхеми генерує завади, які можуть вплинути на точність аналогових вимірювань. Якщо точність перетворення має вирішальне значення, рівень шуму може бути зменшений шляхом застосування таких методів:

1. Лінії аналогових сигналів необхідно робити як можна коротшими. При цьому слід переконатися, що аналогові доріжки знаходяться над аналоговою заземленою поверхнею, і знаходяться як можна далі від високошвидкісних цифрових ліній.

2. Напруга живлення аналогової частини AVCC повинна бути під'єднана до цифрової напруги живлення VCC через LC коло, як показано на рис. 191.

3. Для зменшення шуму від центрального процесора слід використовувати роботу АЦП у режимі сну.

4. Якщо будь-які лінії порту А використовуються як цифрові виходи, важливо, щоб вони не перемикалися під час перетворення АЦП.

6.12.7. Результати перетворення

Опорна напруга для AREF визначає діапазон перетворення для АЦП. Напруга, що перетворюється, не має виходити за межі діапазону, вказаного AREF. При роботі з недиференціальним каналом, якщо напруга на вході АЦП перевищує AREF, результат перетворення дорівнюватиме 0x3FF.

Після завершення перетворення (коли біт ADIF в одиничному стані, а біт ADSC переходить у низький), результат перетворення можна знайти в регістрах результату перетворення АЦП ADCH та ADCL.

Для однополярного перетворення результат перетворення визначається за формулою

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}},$$

де V_{IN} – вхідна напруга, V_{REF} – опорна напруга, ADC – результат перетворення.

Значення 000 представляє напругу, дорівнює і менша нуля, а значення 3FF представляє напругу, що дорівнює $1023 \cdot V_{REF}/1024$ й вищу.

Для диференціального перетворення результат перетворення залежить від включеного коефіцієнта підсилення і визначається за формулою

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot GAIN \cdot 512}{V_{REF}},$$

де V_{POS} – позитивна вхідна напруга, V_{NEG} – негативна вхідна напруга, V_{REF} – опорна напруга, $GAIN$ – коефіцієнт підсилення, ADC – результат перетворення. Результат представляється числом із знаком, значення 200h (-512) відповідає мінімально можливому від’ємному значенню, а значення 1FFh (+511) – максимально можливому позитивному. Тобто старший біт результату у такому випадку є знаковим.

У процесі перетворення АЦП генерує 10-бітовий результат ADC9:0, що представляється кодом у регістрах ADCH і ADCL. За замовчуванням, результат вирівнюється вправо, проте представлення можна змінити за допомогою зміни біту ADLAR у регістрі ADMUX. Вибір типу вирівнювання залежить від завдань, котрі вирішуються. Формат результату перетворення у регістрах ADCH ADCL зображено у наступній таблиці.

Таблиця 89

Вирівнювання результату перетворення

ADLAR	Формат результату перетворення у регістрах ADCH ADCL									
0	Bit	15	14	13	12	11	10	9	8	
								ADC9	ADC8	ADCH
		ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
		7	6	5	4	3	2	1	0	
Вирівнювання вправо										
1	Bit	15	14	13	12	11	10	9	8	
		ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
		ADC1	ADC0	-	-	-	-	-	-	ADCL
		7	6	5	4	3	2	1	0	
Вирівнювання вліво										

У випадку, коли результат вирівняний вліво і є достатньою 8-бітова точність, досить читати регістр ADCH. Якщо ж потрібна більша точність чи результат, вирівняний вправо, необхідно читати регістри у такому порядку:

спочатку прочитати регістр ADCL, а потім – регістр ADCH, щоб гарантувати, що вміст регістрів даних належить до одного й того ж перетворення. Після читання регістра ADCL доступ АЦП до регістрів даних блокується. Це означає, що коли ADCL було прочитано, і перетворення завершується до читання ADCH, жоден регістр не буде оновлено й результат перетворення буде втрачено. Доступ регістрів для АЦП знову вмикається після читання регістра ADCH.

АЦП має своє власне переривання, яке може бути викликане при завершенні перетворення. Коли доступ до регістрів даних для АЦП заборонений між читаннями ADCH і ADCL, переривання буде викликане, навіть якщо результат перетворення був втрачений.

6.12.8. Переривання, пов'язані з АЦП

Із АЦП пов'язане переривання "ADC Conversion Complete", котре викликається при закінченні перетворення АЦП та готовності даних. Адреса підпрограми обробки переривання 020h. Для керування дозволом переривання від АЦП та визначення стану переривання використовуються біти у регістрі ADCSRA (рис. 189).

Біт ADIF (ADC Interrupt Flag) вказує на наявність запиту переривання від АЦП. Біт встановлюється в одиницю, якщо перетворення даних завершено і значення даних у регістрах даних оновлене. Біт скидається апаратно при виклику відповідного переривання, а також може бути скинутий записом одиниці у цей біт.

Біт ADIE (ADC Interrupt Enable) вказує на дозвіл переривання від АЦП. Коли цей біт встановлений в 1 і біт глобального дозволу переривань у регістрі SREG теж встановлений в 1, то при закінченні перетворення буде викликане переривання від АЦП.

6.12.9. Приклади роботи із АЦП

Приклад 1

Використовуючи режим безперервного перетворення, оцифрувати всі 8 вхідних каналів, результати вимірювання записати в пам'ять, починаючи з адреси \$60. Тактова частота процесора 4MHz.

Коефіцієнт ділення частоти має бути 32, оскільки $4000 \text{ кГц}/32 = 125 \text{ кГц}$ і лежить у межах 50 - 200 кГц.

```
Input1:
  ldi R30, $60          ; настроюємо вказівник Z
  ldi R31, 0
  ldi R16, 0b10111101; АЦП увімкнути, Free Run - режим
  out ADCSR, R16       ; коефіцієнт ділення 32
  clr R17               ; встановити нульовий канал
  out ADMUX, R17
  sbi ADCSR, ADSC      ; запустити АЦП
loop:   inc r17         ; встановити наступний канал
  out ADMUX, R17
  sbis ADCSR, ADIF     ; готово?
  rjmp pc-1            ; перехід на попередню команду
  sbi ADCSR ADIF       ; очищаємо ADIF
  in R16, ADCL         ; читаємо і зберігаємо
  st Z+, R16           ; молодший байт
  in R16, ADCH         ; читаємо і зберігаємо
  st Z+, R16           ; старший байт
  cpi R30, $60+7       ; сьоме вимірювання
  brne free
  cbi ADCSR, ADFR      ; вимкнути режим Free Run
free:   cpi R31, $60+8; останнє вимірювання
  brne loop           ; якщо ні - цикл.
ret
```

Приклад 2

Використовуючи Idle-режим, виміряти напругу поточного каналу. Дані зберегти в регістрах R1:R0. Тактова частота мікроконтролера 4 MHz.

```
Input2:
  ldi R16, 0b10011101 ; режим однократного вимірювання
  out ADCSR, R16       ; переривання дозволені
  sei                  ; глобальний дозвіл переривання
  ldi R16, 0b01000000 ; дозвіл Idle-режиму
```

```

out MCUCR, R16
sleep                ; увійти до Idle-режим
in R0, ADCL         ; віднімати молодший байт
in R1,ADCH         ; віднімати старший байт
ret

```

6.13. Компаратор мікроконтролера

Часто при роботі мікро-ЕОМ необхідно порівнювати два аналогових сигнали за рівнем. Для такого порівняння у мікросхемі ATmega32 є блок аналогового компаратора. Аналоговий компаратор порівнює значення вхідної напруги на контакті AIN0 з напругою на контакті AIN1. Структура компаратора наведена на рис. 192.

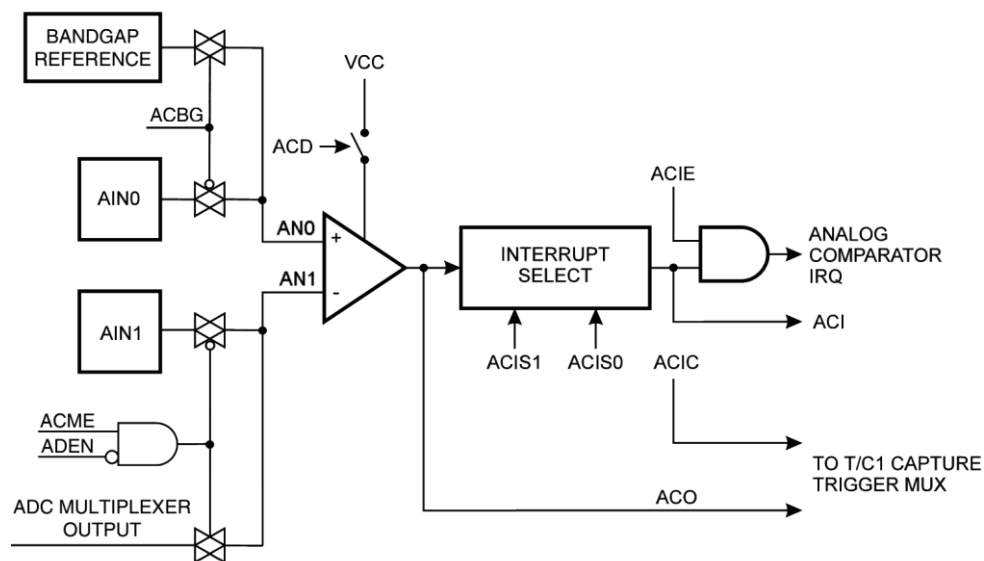


Рис. 192. Будова аналогового компаратора

До структури компаратора входить джерело опорної напруги на основі напруги забороненої зони (Bandgap reference) зі значенням напруги 1,23 В (1,15 В – 1,35 В), два входи AIN0 та AIN1, що можуть бути заблоковані відповідними ключами, власне сам компаратор та блок вибору сигналу переривання (Interrupt select). Позитивний сигнал, що порівнюється, може надходити від входу AIN0 чи від джерела опорної напруги (Bandgap reference) залежно від стану біта ACBG. Негативний сигнал, що порівнюється, може надходити від виводу AIN1 чи від аналогового мультиплектора АЦП (ADC multiplexer output) залежно від стану ліній ACME та ADEN.

Коли напруга на позитивному вході компаратора AN0 є вищою, ніж напруга на негативному вході AN1, вихід компаратора АСО встановлюється в одиничний стан. Вихід компаратора може бути встановлений, щоб викликати функцію захоплення таймера-лічильника 1 (T/C1 capture trigger). Крім того, компаратор може викликати окреме переривання виключно для аналогового компаратора (Analog comparator IRQ). За допомогою блоку вибору сигналу переривання (Interrupt select) програміст може обрати формування переривання при перевищенні вхідного сигналу на лінії AN0 над AN1, при зворотному перевищенні рівня AN1 відносно AN0 та при будь-якому перемиканні стану виходу компаратора.

6.13.1. Вхідні сигнали компаратора

Як вже відзначалося, у якості позитивного вхідного сигналу компаратора AN0 може бути обрано або аналоговий сигнал з лінії AIN0, або опорний сигнал напруги забороненої зони (Bandgap reference 1,23 В). Вибір джерела позитивного аналогового сигналу здійснюється бітом ACBG у регістрі ACSR (Analog Comparator Control and Status Register). Одиничне значення біту вказує на ввімкнення на вхід напруги забороненої зони, нульове – на під'єднання на вхід AN0 лінії AIN0.

В якості негативного вхідного сигналу компаратора AN1 може бути обрано або аналоговий сигнал з лінії AIN1, або сигнал ADC Multiplexer output з виходу мультиплексора позитивного аналогового сигналу АЦП (Pos. input mux) (див. рис. 185). Сигнал із позитивного входу аналогового мультиплексора буде передаватися на вхід AN1 лише при комбінації бітів: ACME = 1 та ADEN = 0. Біт ACME знаходиться у регістрі SFIOR, а біт ADEN є бітом ввімкнення/вимкнення АЦП, який знаходиться у регістрі ADCSRA (ADC Control and Status Register A). Нульове значення біту ADEN вказує на вимкнення АЦП. Отже, для використання аналогового мультиплексора разом із компаратором необхідно, щоб АЦП мікросхеми був вимкнений.

Таблиця 90

Вибір від'ємного входу аналогового компаратора

АСМЕ	ADEN	MUX2..0	Негативний вхід AN1 аналогово компаратора
0	x	xxx	AIN1
1	1	xxx	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

Вибір потрібного від'ємного входу аналогового компаратора здійснюється згідно з таблицею 90. Біт АСМЕ знаходиться у регістрі SFIOR, біт ADEN – у регістрі ADCSRA, біти MUX2..0 знаходяться у регістрі ADMUX.

6.13.2. Переривання від аналогового компаратора

Із аналоговим компаратором пов'язане переривання " Analog Comparator Interrupt", котре викликається або при перевищенні рівня сигналу AN0 над AN1, або при зворотному перевищенні, або при будь-якій зміні співвідношення між сигналами AN0 над AN1. Адреса підпрограми обробки переривання 024h.

Для керування дозволом переривання від компаратора та визначення стану переривання використовуються біти у регістрі ACSR (Analog Comparator Control and Status Register).

Біт ACI (Analog Comparator Interrupt Flag) вказує на наявність запиту переривання від аналогового компаратора. Біт встановлюється в одиницю, якщо запрограмована подія, що задеться бітами ACIS1 та ACIS0, відбулася. Біт скидається апаратно при виклику відповідного переривання, а також може бути скинений записом одиниці у цей біт.

Біт ACIE (Analog Comparator Interrupt Enable) вказує на дозвіл переривання від компаратора. Коли цей біт встановлений в 1 і біт глобального дозволу переривань у регістрі SREG теж встановлений в 1, при запрограмованій події буде відбуватися переривання від компаратора.

Біти ACIS1 та ACIS0 (Analog Comparator Interrupt Mode Select) задають тип події, при котрій відбувається запит переривання. При зміні стану бітів переривання від компаратора мають бути заблоковані, інакше може виникнути запит переривання у момент зміни стану бітів ACIS1 та ACIS0. Події, при котрих викликається переривання від аналогового компаратора, занесені у таблицю 91.

Таблиця 91

Події, при котрих викликається переривання від компаратора

ACIS1:0	Подія, при котрій викликається переривання
0 0	Будь-яка зміна співвідношення між входами AN0 та AN1
01	Зарезервовано
1 0	Сигнал на лінії AN1 перевищив стан лінії AN0
1 1	Сигнал на лінії AN0 перевищив стан лінії AN1

6.13.3. Регістри контролю аналогового компаратора

Для контролю стану компаратора та задавання його режиму роботи використовують такі регістри:

- ◇ ACSR (Analog Comparator Control and Status Register) – регістр контролю та статусу аналогового компаратора;
- ◇ ADC multiplexer select (ADMUX) – регістр вибору вхідного мультиплексора, що може використовуватися для вибору негативного входу аналогового компаратора;
- ◇ ADC ctrl. & status register (ADCSRA) – регістр контролю та статусу АЦП, із котрого може використовуватися біт ADEN для вибору негативного входу аналогового компаратора;
- ◇ SFIOR Special FunctionIO Register – регістр спеціальних функцій вводу-виводу, у котрому використовується біт ACME.

Формат регістра SFIOR наведений на наступному рисунку.

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Рис. 193. Формат регістра SFIOR

Із компаратором пов'язаний біт ACME: (Analog Comparator Multiplexer Enable) – дозвіл роботи аналогового мультиплексора із аналоговим компаратором. Коли біт встановлений в 1 та АЦП вимкнений (біт ADEN у регістрі ADCSRA дорівнює 0), мультиплексор вибирає сигнал на потрібний негативний вхід AN1 для аналогового компаратора з ліній AD0:7. Коли біт встановлений у нуль, на негативний вхід аналогового компаратора надходить сигнал з лінії AIN1.

Bit	7	6	5	4	3	2	1	0	
	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

Рис. 194. Формат регістра ACSR

Формат регістра ACSR (Analog Comparator Control and Status Register) – регістр контролю та статусу аналогового компаратора, наведений на рис. 194. У регістрі біти мають таке призначення:

Біт 7 – ACD (Analog Comparator Disable) – увімкнення та вимкнення аналогового компаратора. Встановлення біту в одиничний стан вимикає аналоговий компаратор, встановлення біту у нульовий стан вмикає його. Зміна біту ACD потребує також заборону переривання від аналогового компаратора, інакше можлива генерація переривань при зміні стану біту ACD.

Біт 6 – ACBG (Analog Comparator Bandgap Select) – вибору джерела позитивного сигналу для аналогового компаратора. Одиничне значення біту вказує на увімкнення на вхід напруги забороненої зони (напруга 1,23 В), нульове – на під'єднання на вхід AN0 лінії AIN0.

Біт 5 – ACO (Analog Comparator Output) – виходу аналогового компаратора. Одиничне значення біту вказує, що сигнал на вході AN0 перевищує значення AN1, а нульове значення вказує, що сигнал на вході AN1 перевищує значення AN0. Стан біту регістра синхронізується з сигналом на виході компаратора із затримкою в 1–2 такти тактової частоти процесора.

Біт 4 – ACI (Analog Comparator Interrupt Flag) – переривання від аналогового компаратора. Біт вказує на наявність запиту переривання від

аналогового компаратора. Біт встановлюється в одиницю, якщо запрограмована подія, задана бітами ACIS1 та ACIS0, відбулася. Біт скидається апаратно при виклику відповідного переривання, а також може бути скинутий записом одиниці у цей біт.

Біт 3 – ACIE (Analog Comparator Interrupt Enable) – вказує на дозвіл переривання від компаратора. Коли цей біт встановлений в 1 і біт глобального дозволу переривань у регістрі SREG теж встановлений в 1, при запрограмованій події буде відбуватися переривання від компаратора.

Біт 2 – ACIC (Analog Comparator Input Capture Enable) – Біт дозволу захоплення від аналогового компаратора. Коли біт знаходиться в одиничному стані, вмикається функція захоплення у таймері-лічильнику 1, а вихід компаратора під'єднується до відповідного входу таймера-лічильника 1. Встановлення біту в нульовий стан розриває зв'язок між таймером-лічильником 1 та аналоговим компаратором.

Біти 1 та 0 – ACIS1 та ACIS0 (Analog Comparator Interrupt Mode Select) задають тип події, при котрій відбувається запит переривання. При зміні стану бітів переривання від компаратора мають бути заблоковані, інакше може виникнути запит переривання у момент зміни стану бітів ACIS1 та ACIS0. Події, при котрих викликається переривання від аналогового компаратора, занесені у таблицю 91.

Формат регістрів ADCSRA та ADMUX наведений на рис. 189 та рис. 190 відповідно.

6.14. Контрольні запитання

1. Основні технічні характеристики мікроконтролерів.
2. Система команд.
3. Проектування мікропроцесорних систем на базі ATmega32.
4. Контролер ATmega32. Основні характеристики.
5. Архітектура контролера ATmega32. Регістри контролера, їх призначення. Регістр стану.
6. Організація резидентної пам'яті даних та пам'яті програм ATmega32.
7. Система переривань.
8. Переривання від таймерів.
9. Переривання від портів.
10. Переривання від EEPROM.
11. Стек і повернення з підпрограм.
12. Порти вводу-виводу.
13. Регістри портів.
14. Схема ліній портів, конфігурування портів на ввід чи вивід.

СПИСОК ЛІТЕРАТУРИ

1. Сташин, В.В. Проектирование цифровых устройств на однокристалльных микроконтролерах [Текст] / В.В. Сташин, А.В. Урсов, О.Ф. Мологонцева. – М.: Энергоатомиздат, 1990. – 224 с.
2. Предко, М. Руководство по микроконтролерам [Текст] / М. Предко. – М.: Постмаркет, 2001.
3. Зубчук, В.И. Справочник по цифровой схемотехнике [Текст] / В.И. Зубчук, В.П. Сигорский, АН. Шкуро. – К.: Техніка, 1990. – 448 с.
4. Проектирование микропроцессорной электронно-вычислительной аппаратуры: справочник [Текст] / В.Г. Артюхов, А.А. Будняк, В.Ю. Лапий, С.М. Молявко, А.И. Петренко. – К.: Техніка, 1988. – 263 с.
5. Микропроцессорный комплект К1810. Структура, программирование, применение [Текст] / Ю.М. Казаринов, В.Н. Номоконов, Г.С. Подклетнов, Ф.В. Филиппов. – М.: Высшая школа, 1990. – 269 с.
6. Мікропроцесорна техніка [Текст] / Ю.І. Якименко, Т.О. Терещенко, Є.І. Сокол, В.Я. Жуйков, Ю.С. Петергера; за ред. Терещенко Т.О. – К.: Кондор, 2004. – 440 с.
7. Бродин, В.Б. Системы на микроконтролерах и БИС программируемой логики [Текст] / В.Б. Бродин, А.В. Калинин. – М.: ЭКОМ, 2002. – 400 с.
8. Коффрон, Дж. Технические средства микропроцессорных систем: практический курс [Текст] / Дж. Коффрон; пер. с англ. Балыбердина А.В. – М.: Мир, 1983. – 344 с.
9. Однокристалльные микроЭВМ [Текст] / А.В. Боборыкин, Г.П. Липовецкий, Г.В. Литвинский, О.Н. Оксинь, С.В. Прохорчик, Л.В. Проценко, Н.В. Пертенко, А.А. Сергеев, П.В. Сивобород. – М.: МИКАП, 1994. – 400 с.
10. Ульрих, В.А. Микроконтролеры PIC16X7XX [Текст] / В.А. Ульрих. – 2-е изд., перераб и доп. – СПб.: Наука и техника, 2002. – 320 с.

11. Тавернье, К. PIC-Микроконтролер. Практика применения [Текст] / К. Тавернье. – М.: ДМКПресс, 2004. – 272 с.
12. 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>.
13. 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>.
14. Евстифеев, В.А. Микроконтроллеры AVR семейства Mega. Руководство пользователя [Текст] / В.А. Евстифеев. – М.: Изд. дом "Додека - XXI", 2007.

Навчально-методична література

В.Р. Медвідь, В.П. Пісьціо, І.Р. Козбур

Навчальний посібник

ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНИХ СИСТИМ КЕРУВАННЯ

Редактор *Є. І. Грищенко*

Коректор *М. Д. Радик*

Комп'ютерне верстання *А.П.Катрич*

Формат 60x90/16. Обл. вид. арк. 23,08. Тираж 300 пр. Зам. № 2488.

Видавництво Тернопільського національного
технічного університету імені Івана Пулюя.

46001, м. Тернопіль, вул. Руська, 56.

Свідоцтво суб'єкта видавничої справи ДК № 4226 від 08.12.11.