

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА
СПОРТУ УКРАЇНИ**

Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

лабораторних робіт

з дисципліни «Вбудовані комп'ютерні системи»
для студентів усіх форм навчання спеціальностей
8.05010203 «Спеціалізовані комп'ютерні системи»
кафедри «Комп'ютерні системи і мережі»

2012

Методичні вказівки до лабораторних робіт з дисципліни «Вбудовані комп'ютерні системи» для студентів усіх форм навчання спеціальностей 8.05010203 «Спеціалізовані комп'ютерні системи» кафедри «Комп'ютерні системи і мережі» / Укладачі: Проскурін М.П., Грушко С.С.– Запоріжжя: ЗНТУ, 2012.– 86 с.

Укладачі: М.П. Проскурін, доцент, к.т.н.;
С.С. Грушко, асистент.

Рецензент: А. І. Вершина, доцент, к.т.н.

Відповідальний за випуск: С.С. Грушко, асистент

Затверджено
на засіданні кафедри
«Комп'ютерні системи і мережі»
Протокол № 7
від «11» травня 2012 р.

ЗМІСТ

ВСТУП	5
1 ОСОБЛИВОСТІ МІКРОКОНТРОЛЕРІВ СЕРІЇ МК51.....	7
1.1 Відомості про однокристальні 8-розрядні мікроконтролери сімейства МК51, їх характеристика	7
1.2 Про деякі особливості функціонування МК51.....	13
1.3 Функціональна схема включення МК51 із зовнішнім ППЗП програм.....	13
1.4 Арифметико-логічний пристрій (АЛП). Регістр PSW.....	21
1.5 Блок таймерів/лічильників. Регістри TMOD і TCON.....	23
1.6 Блок послідовного інтерфейсу і переривань. Регістри SCON, IP, IE	32
1.7 Порти	37
1.8 Пам'ять даних	38
1.9 Пам'ять програм	41
1.10 Блок керування. Синхронізація МК. Регістр PCON. Режими зменшеного енергоспоживання.....	44
1.11 Система команд МК51	52
2 ЛАБОРАТОРНА РОБОТА №1	55
2.1 Порядок виконання лабораторної роботи	55
2.2 Короткі теоретичні відомості.....	55
2.3 Параметри ЦАП	57
2.4 Приклад виконання програми цифро-аналогового перетворення.....	58
2.5 Завдання до лабораторної роботи	59
2.6 Контрольні питання	60
3 ЛАБОРАТОРНА РОБОТА №2	62
3.1 Порядок виконання лабораторної роботи:	62
3.2 Короткі теоретичні відомості.....	62
3.3 Приклад реалізації алгоритму порозрядного врівноваження	64
3.4 Завдання до лабораторної роботи	66
3.5 Контрольні питання	66
4 ЛАБОРАТОРНА РОБОТА № 3	68
4.1 Короткі теоретичні відомості.....	68
4.1.1 Підрахунок числа імпульсів між двома подіями	68
4.1.2 Підрахунок числа імпульсів за заданий проміжок часу....	69
4.1.3 Формування вихідних динамічних керуючих впливів.....	70

4.1.4 Формування вихідних статичних сигналів керування	71
4.2 Завдання до лабораторної роботи	73
4.3 Порядок виконання лабораторних робіт із застосуванням середовища «Proteus» (demo- версія)	73
5 ЛАБОРАТОРНА РОБОТА № 4	74
5.1 Короткі теоретичні відомості.....	74
5.2 Завдання до лабораторної роботи	81
Додаток 1 Арифметичні операції	84

ВСТУП

Дисципліна «Вбудовані комп'ютерні системи» (ВКС) є однією з завершальних в системі знань і вмінь, що до використання мікропроцесорів (МП), мікроконтролерів (МК) у вигляді цифрових керуючих пристроїв на їх основі – цифрових автоматів (ЦА), для вирішення різноманітних завдань в галузях науки, техніки, інформаційних технологій і виробництва.

Курс лабораторних робіт по ВКС має за ціль сформувати навички фахівця-інженера системотехніки (спеціаліста, магістра) за спеціальностями «Спеціалізовані комп'ютерні системи» та «Комп'ютерні системи та мережі» при вирішенні ним конкретних типових завдань з використанням МП, МК.

Вивчення дисципліни «Вбудовані комп'ютерні системи» надає студентам можливість отримати теоретичну підготовку для складання уявної і реальної моделі простих цифрових керуючих пристроїв на основі МП і МК, ознайомитися з їх структурними і електричними схемами, різновидами діаграм процесів обміну сигналами, інтерфейсів; вчитися розробляти і аналізувати алгоритми отримання, кодування (декодування), зберігання і переробки потоків дискретних даних, що несуть інформацію про протікання будь-яких процесів в галузі науки, техніки і технологій (стан параметрів: електродвигуна, турбіни, робота, системи обробки зображень, автоматизованого комплексу, ін.).

Метою запропонованого курсу лабораторних робіт (ЛР) з дисципліни (ВКС) є практичне вивчення методів, підходів для розвитку і формування у студентів навичок розробки та налагодження програм для побудови найпростіших цифрових керуючих пристроїв на базі відомих типів МП, МК згідно з отриманим варіантним завданням. Основою для проведення курсу ЛР і макетування роботи розроблених пристроїв є апаратно-програмний стенд EV8031/AVR.

Виконання студентами ЛР з курсу ВКС дозволяє їм ознайомитись з основними етапами формування технічного завдання (ТЗ) на розробку цифрового керуючого пристрою, скласти модель його функціонування і блок схему, провести макетування або

модельовання створеного ЦА, перевірити та закріпити отримані теоретичні знання і освоїти практичні навички.

В результаті проведення і захисту ЛР **студенти повинні знати:** основні засоби перетворення і прийоми отримання, зберігання, обробки аналогових, цифрових інформаційних потоків (на основі МП, МК); формування технічного завдання (ТЗ) по створення конкретного пристрою; згідно вимог ТЗ проводити вибір необхідного МП, МК та його периферійних пристроїв (генераторів, цифро-аналогових і аналого-цифрових перетворювачів, ін.); **вміти розробляти** найпростіші схеми, алгоритми і налагоджувати програми для них; використовувати для відлагодження віртуальних моделей пристроїв програмно-модельюючі середовища типу Proteus, MultiSim, ін..

1 ОСОБЛИВОСТІ МІКРОКОНТРОЛЕРІВ СЕРІЇ МК51

1.1 Відомості про однокристалні 8-розрядні мікроконтролери сімейства МК51, їх характеристика

Сімейство однокристалних 8-розрядних мікроконтролерів (МК) серій 1816, 1830 включає ряд моделей, модифікації яких варіюються залежно від обсягу і характеру обчислювальних ресурсів (пам'яті програм і даних, тактової частоти).

Сімейство вітчизняних МК51 включає 5 основних модифікацій серій КР1816, КР1830, що відрізняються по реалізації резидентної пам'яті програм і потужності споживанні (Мікросхема КР1835ВЕ51* є одним з модифікованих варіантів КР1830ВЕ51).

У таблиці 1.1 наведено основні моделі із зазначеного сімейства МК (далі МК51).

Таблиця 1.1 - Сімейство МК51 (МСС51)

Модель	Аналог	Обсяг резидентної пам'яті програм, Кбайт	Обсяг резидентної пам'яті даних, байт	Тактова частота, МГц	Струм споживання, мА
КР1816ВЕ31	8031АН (8031)	зовнішня	128	12	150
КР18І6ВЕ51	8051АН (8051)	4 К	128	12	150
КМ1816ВЕ751	875ІАН (8751)	4 К	128	12	220
КР1830ВЕ31	80С3ІВН (80С31)	зовнішня	128	12	18
КР1830ВЕ51	80С51ВН (80С51)	4 К	128	12	18
КР1835ВЕ51*	80С51ВU (80С51)	4 К	128	12	18

З таблиці 1.1 видно, що найбільш економічними є великі інтегральні схеми (BIC) серії KP1830 при однакових значеннях основних технічних характеристик.

Їхнім недоліком є менший температурний діапазон експлуатації (-10 - +70 С), у той час як закордонний фірми Intel, Siemens роблять модифікації контролерів, розраховані на застосування в діапазонах: 0 - +70 С, -40 - +85 С, -40 - +110 С і -40 - +125С [5].

KM1816BE751, у відмінності від KP1816BE51, містить внутрішнє ППЗП ємністю 4 Кбайт із ультрафіолетовим стиранням.

Слід зазначити, що у всіх моделей МК51 за рахунок використання зовнішньої пам'яті, ємність пам'яті програм і даних може бути розширена до 64 Кбайт.

Приведемо деякі особливості моделей KM1830BE751 і KM1830BE753 (аналог 8753Н фірми AMD). Остання відрізняється наявністю перепрограмувального запам'ятовувального пристрою (ППЗП) з ультрафіолетовим стиранням на 8 Кбайт.

Мікросхеми мають особливості:

- спеціальний режим експлуатації;
- додаткові засоби захисту пам'яті програм, розташованої на кристалі, два біти захисту пам'яті і шифрувальну таблицю;
- алгоритм програмування вкороченими імпульсами.

Восьмирозрядні однокристалні мікроконтролери сімейства МК51 виконані по n-МОП технології (серія 1816) і КМОП технології (серія 1830).

Кожний МК розглянутого сімейства містить вбудоване ОЗП пам'яті даних ємністю 128 байт із можливістю розширення загального обсягу оперативної пам'яті даних до 64 Кбайт за рахунок використання зовнішніх мікросхем ЗППВ.

Загальний обсяг пам'яті МК сімейства МК51 може досягати 128 Кбайт: 64 Кбайт – пам'ять програм і 64 Кбайт – пам'ять даних.

При розробці на базі МК більш складних систем можуть бути використані стандартні ІС із байтової організацією, наприклад, серії KP580.

МК містять усі вузли, необхідні для автономної роботи:

- центральний восьмирозрядний процесор;
- пам'ять програм обсягом 4 Кбайт (тільки KM1816BE751, KP1816BE51 і KP1830BE51);
- пам'ять даних обсягом 128 байт;

- чотири восьмирозрядних програмувальних каналів введення-виводу (порти P0, P1, P2, P3);
- два 16-бітових багаторежимних таймера/лічильника;
- система переривань із п'ятьома векторами і двома рівнями;
- послідовний інтерфейс;
- тактовий генератор.

Система переривань, блок послідовного інтерфейсу і таймери об'єднані в один блок.

Використання мікроконтролера сімейства МК51 у порівнянні із МК48 забезпечує збільшення обсягу пам'яті команд і пам'яті даних. Нові можливості введення-виводу і периферійних пристроїв розширюють діапазон застосування і знижують загальні витрати системи. Залежно від умов використання швидкодія системи збільшується мінімум у два з половиною рази і максимум вдесятеро.

МК КМ1816ВЕ751 містить ППЗП ємністю 4096 байт зі стиранням ультрафіолетовим випромінюванням і зручний на етапі розробки системи при налагодженні програм, а також при виробництві невеликими партіями або при створенні систем, що вимагають у процесі експлуатації періодичного підстроювання. За рахунок використання зовнішніх мікросхем пам'яті загальний обсяг пам'яті програм може бути розширений до 64 Кбайт.

МК КР1816ВЕ31 і КР1830ВЕ31 не містять вбудованої пам'яті програм і можуть використовувати до 64 Кбайт зовнішньої постійної (або перепрограмувальної пам'яті програм) та ефективно використовуватися в системах, що вимагають суттєво більшого за обсягом ПЗП пам'яті програм (на кристалі тільки 4 Кбайт).

МК51 має: пам'ять програм у вигляді постійного запам'ятовуючого пристрою (ПЗП): призначена для зберігання програм і має окреме від пам'яті даних адресний простір обсягом до 64 Кбайт, причому для мікросхем КР1816ВЕ51, КМ1816ВЕ751 і для КР1830ВЕ51 частина пам'яті програм з адресами 0000Н -0FFFFН розташована на кристалі МК. Пам'ять програм, розташована на кристалі, складається з 12-розрядного дешифратора і ПЗП ємністю 4 К*8 біт для мікросхем КР1816ВЕ51, КР1830ВЕ51 або перепрограмуємого ПЗП (ППЗП) з ультрафіолетовим стиранням ємністю 4 К*8 біт для КМ1816ВЕ751. Запис програм у ПЗП відбувається під час виготовлення кристалів.

Регістри особливого призначення (РОП): МК має 32 РОП;
Програмно-керовані «прапори» (ПКП): МК має 128
 обумовлених користувачем;

Набір реєстрів спеціальних функцій (РСФ):

Таблиця 1.2 – Набір реєстрів спеціальних функцій (РСФ)

Позначення	Найменування	Адреса
* ACC	Акумулятор	0E0H
* B	Регістр В	0F0H
* PSW	Регістр стану програми	0D0H
SP	Показчик стека	81H
DPTR	Показчик даних. 2 байта:	
DPL	Молодший байт	82H
DPH	Старший байт	83H
* P0	Порт 0	80H
* P1	Порт 1	90H
* P2	Порт 2	0A0H
* P3	Порт 3	0B0H
* IP	Регістр пріоритетів переривань	0B8H
* IE	Регістр дозволу переривань	0A8H
TMOD	Регістр режимів таймера/лічильника	89H
* TCON	Регістр управління таймера/лічильника	88H
TH0	Таймер/лічильник 0. Старший байт	8CH
TL0	Таймер/лічильник 0. Молодший байт	8AH
TH1	Таймер/лічильник 1. Старший байт	8DH
TL1	Таймер/лічильник 1. Молодший байт	8BH
* SCON	Керування послідовним портом	98H
SBUF	Буфер послідовного порту	99H
PCON	Керування споживанням	87H

* - реєстри, що допускають побітову адресацію.

РОП і обумовлені користувачем ПКП розташовані в адресному просторі **внутрішнього** ОЗП даних (ВОЗП). Регістри спеціальних

функцій (SFR, SPECIAL FUNCTION REGISTERS) із вказівкою їх адрес наведено в таблиці 1.2

Нижче коротко описуються функції регістрів, наведених у таблиці 1.2. Докладно ці регістри розглядаються у відповідних розділах справжнього опису.

Акумулятор. ACC — регістр акумулятора. Команди, призначені для роботи з акумулятором, використовують мнемоніку "A", наприклад, MOV A, P2. Мнемоніка "ACC" використовується, приміром, при побітовій адресації акумулятора. Так, символічне ім'я п'ятого біта акумулятора при використанні асемблера ASM51 буде наступним: ACC.5.

Регістр В. Використовується під час операцій множення і ділення. Для інших інструкцій регістру може розглядатися як додатковий надшвидкий регістр.

Регістр стану програми. Регістр PSW містить інформацію про стан програми.

Показчик стека SP. 8-бітовий регістр, уміст якого інкрементується перед записом даних у стек при виконанні команд PUSH і CALL. При початковому скиданні показчик стека встановлюється в 07H, а область стека в ОЗП даних починається з адреси 08H. При необхідності шляхом перевизначення показчика стека область стека може бути розташована в будь-якій місці внутрішнього ОЗП даних МК.

Показчик даних. Показчик даних (DPTR) складається зі старшого байта (DPH) і молодшого байта (DPL). Містить 16-бітову адресу при звертанні до зовнішньої пам'яті. Може використовуватися як 16-бітовий регістр або як два незалежні восьми бітові регістри.

Порт0 -Порт3. Регістрами спеціальних функцій P0, P1, P2, P3 є регістри-"засувки" відповідно портів P0, P1, P2, P3.

Буфер послідовного порту. SBUF являє собою два окремі регістри: буфер передавача і буфер приймача. Коли дані записуються в SBUF, вони надходять у буфер передавача, причому запис байта в SBUF автоматично ініціює його передачу через послідовний порт. Коли дані читаються з SBUF, вони вибираються з буфера приймача.

Регістри таймера. Регістрові пари (TH0.TL0) і (TH1.TL1) утворюють 16-бітові рахункові регістри відповідно таймера/лічильника 0 і таймера/лічильника 1.

Регістри керування. Регістри спеціальних функцій IP, IE, TMOD, TCON, SCON і PCON містять біти керування і біти стану системи переривань, таймерів/лічильників і послідовного порту. МК при функціонуванні забезпечує:

- мінімальний час виконання команд додавання - 1 мкс;
- апаратне множення/ділення з мінімальним часом виконання команд множення/ділення - 4 мкс.

В МК передбачена можливість завдання частоти внутрішнього генератора за допомогою кварцу, LC - ланцюжка або зовнішнього генератора.

Архітектура сімейства МК51 незважаючи на те, що вона заснована на архітектурі сімейства МК48, все-таки не є повністю сумісною з нею. У новім сімействі є ряд нових режимів адресації, додаткові інструкції, розширений адресний простір і ряд інших апаратних відмінностей. Розширена система команд забезпечує побайтову і побітову адресацію, двійкову і двоїчно-десяткову арифметику, індикацію переповнення і визначення парності/непарності, можливість реалізації логічного процесора. Найважливішою і відмітною рисою архітектури сімейства МК51 є те, що АЛП може, поряд з виконанням операцій над 8-розрядними типами даних, маніпулювати однорозрядними даними. Окремі програмно-доступні біти можуть бути встановлені, скинуті або замінені їхнім доповненням, можуть пересилатися, перевірятися і використовуватися в логічних обчисленнях. Тоді як підтримка простих типів даних (при існуючій тенденції до збільшення довжини слова) може з першого погляду здатися кроком назад, ця якість робить МК сімейства МК51 особливо зручними для застосувань, у яких використовуються контролери. Алгоритми роботи останніх по своїй суті припускають наявність вхідних і вихідних булевих змінних, які складно реалізувати за допомогою стандартних мікропроцесорів. Усі ці властивості в цілому називаються булевым процесором сімейства МК51. Завдяки такому потужному АЛП набір інструкцій МК сімейства МК51 однаково добре підходить як для застосувань керування в реальному масштабі часу, так і для алгоритмів з більшим обсягом даних.

1.2 Про деякі особливості функціонування МК51

Якщо на вивід МК EA/NPP подана напруга живлення Vcc то звертання до зовнішньої пам'яті програм відбувається автоматично при виробітку лічильником команд адреси, що перевищує 0FFFFH.

Якщо адреса перебуває в межах 0000H - 0FFFFH, обіг відбувається до пам'яті програм, розташованої на кристалі (внутрішньої пам'яті програм).

Якщо на вивід EA/NPP поданий «0», внутрішня пам'ять програм відключається і, починаючи з адреси 0000H, усі звернення виконуються до зовнішньої пам'яті програм.

Якщо МК не має внутрішньої пам'яті програм, її вивід EA/NPP повинен бути підключений до загальної шини.

Читання із зовнішньої пам'яті програм стробується сигналом PSEN. При роботі із внутрішньою пам'яттю програм сигнал PSEN не формується. МК не мають інструкцій і апаратних засобів для програмного запису на згадку програм.

При звертаннях до зовнішньої пам'яті програм завжди формується 16-розрядна адреса, молодший байт якого видається через порт P0, а старший - через порт P2. При цьому байт адреси, видаваний через порт P0, повинен бути зафіксований у зовнішньому регістрі по спаду сигналу АЛП ALE, тому що надалі лінії порту P0 використовуються в якості шини даних, по якій байт із зовнішньої пам'яті програм вводиться в МК.

1.3 Функціональна схема включення МК51 із зовнішнім ППЗП програм

Порт P0 працює як мультиплексорна шина адреса/даних: видає молодший байт лічильника команд, а потім переходить у високоімпедансне становище і очікує приходу байта із ППЗП програм. Коли молодший байт адреси перебуває на виходах порту P0, сигнал ALE залишає його в адресному регістрі RG. Старший байт адреси перебуває на виходах порту P2 протягом усього часу звертання до ППЗП. Сигнал PME дозволяє вибірку байта із ППЗП, після чого обраний байт надходить на порт P0 МК51 і вводиться в МК.

Пам'ять даних призначена для приймання, зберігання і видачі інформації, використаної в процесі виконання програми. Пам'ять даних, розташована на кристалі МК, складається з регістру адреси ОЗП, дешифратора, ОЗП і покажчика стека.

Регістр адреси ОЗП призначений для приймання і зберігання адреси обраної за допомогою дешифратора комірки пам'яті, яка може містити як біт, так і байт інформації.

ОЗП являє собою 128 восьмирозрядних регістрів, призначених для приймання, зберігання і видачі різної інформації.

Показчик стека являє собою восьмирозрядний регістр, призначений для приймання і зберігання адреси гнізда стека, до якої було останнє звернення. При виконанні команд LCALL, ACALL вміст показчика стека збільшується на 2. При виконанні команд RET, RETI вміст показчика стека зменшується на 2. При виконанні команди PUSH direct вміст показчика стека збільшується на 1. При виконанні команди POP direct вміст показчика стека зменшується на 1. Після скидання в показчику стека встановлюється адреса 07H, що відповідає початку стека з адресою 08H.

У МК передбачена можливість розширення пам'яті даних шляхом підключення зовнішніх пристроїв ємністю до 64 Кбайт. При цьому звертання до зовнішньої пам'яті даних можливо тільки за допомогою команд MOVX.

Команди MOVX @Ri, A і MOVX A, @Ri формують восьмирозрядну адресу, видану через порт P0. Команди MOVX @DPTR, A і MOVX @A, DPTR формують 16-розрядну адресу, молодший байт якої видається через порт P0, а старший – через порт P2.

Байт адреси, виданий через порт P0, повинен бути зафіксований у зовнішньому регістрі по спаду сигналу АЛП ALE, тому що надалі лінії порту P0 використовуються як шина даних, через яку байт даних приймається з пам'яті при читанні або видається в пам'ять даних при записі. При цьому читання стробується сигналом RD, а запис – сигналом WR. При роботі із внутрішньою пам'яттю даних сигнали RD і WR не формуються.

Порти P0, P1, P2, P3 є двонаправлені порти введення-виводу і призначені для забезпечення обміну інформацією МК із зовнішніми пристроями, утворюючи 32 лінії вводу-виводу. Кожний з портів містить фіксатор-засувку, який являє собою восьмирозрядний регістр, що має байтову і бітову адресацію для установки (скидання) розрядів за допомогою програмного забезпечення.

Фізичні адреси P0, P1, P2, P3 становлять для:

- P0 – 80H, при бітовій адресації 80H -87H; P2 – A0H, при бітовій адресації A0H – A7H;
- P1 – 90H, при бітовій адресації 90H -97H; P3 – B0H, при бітовій адресації B0H – B7H.

Крім роботи як звичайних портів введення/виводу лінії портів P0 – P3 можуть виконувати ряд додаткових функцій, описаних нижче.

Структура мікроконтролера KP1816BE51 (МК51):

Умове графічне позначення мікроконтролера МК51 показано на рис. 1.1.

МК51 працює від внутрішнього генератора, який має зовнішні виводи для підключення кварцового резонатора.

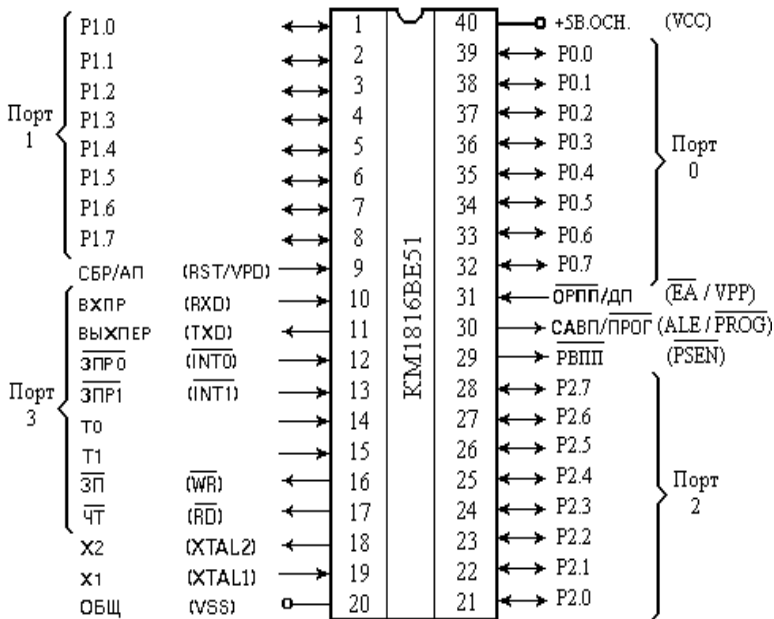


Рисунок 1.1 – Умове графічне позначення мікроконтролера МК51

Призначення виводів наведено в таблиці 1.3:

Uss - потенціал загального дрота ("землі");

Ucc - основна напруга живлення +5 В;

X1, X2 - виводи для підключення кварцового резонатора;

RST/VPD - вхід загального скидання/підключення резервного джерела живлення;

PSEN - дозвіл зовнішньої пам'яті програм; видається тільки при звертанні до зовнішнього ПЗП;

ALE - строб адреси зовнішньої пам'яті;

EA - відключення внутрішньої програмної пам'яті; рівень 0 на цьому вході змушує мікроконтролер виконувати програму тільки зовнішньої ПЗП; ігноруючи внутрішню (якщо остання є);

P0 - восьмибітний двонаправлений порт введення-виводу інформації: при роботі із зовнішніми ОЗП і ПЗП по лініях порту в режимі тимчасового мультиплексування видається адреса зовнішньої пам'яті, після чого здійснюється передача або приймання даних;

P1 - восьмибітний квазідвонаправлений порт введення/виводу: кожний розряд порту може бути запрограмований як на введення, так і на вивід інформації, незалежно від стану інших розрядів;

P2 - восьмибітний квазідвонаправлений порт, аналогічний P1; крім того, виводи цього порту використовуються для видачі адресної інформації (старшого байта) при звертанні до зовнішньої пам'яті програм або даних (якщо використовується 16-бітова адресація останньої). Виводи порту використовуються так само для підключення і програмування розширника (K580BP43);

P3 - восьмибітний квазідвонаправлений порт, аналогічний P1; крім того, виводи цього порту можуть виконувати ряд альтернативних функцій, які використовуються при роботі таймерів, порту послідовного введення-виводу, контролера переривань і зовнішньої пам'яті програм і даних.

Альтернативні функції порту P3:

Rxd – вхід приймача послідовного порту в режимі УАПП. Введення/вивід даних у режимі регістру, що зрушує;

Txd – вихід передавача послідовного порту в режимі УАПП. Вихід синхронізації в режимі регістру, що зрушує;

INT0 (інв), INT1 (інв) – входи запитів на переривання (по низькому рівню або зрізі);

TO, T1 – входи таймерів-лічильників або тест-входи;

RD (інв), WR (інв) – сигнали керування читання і запису (формується при звертанні до зовнішньої пам'яті).

Таблиця 1. 3- Призначення виводів МК51

№ виводу	Позначення	Призначення	Тип
1-8	P1.0-P1.7	8-розрядний двонаправлений порт P1. Вхід адреси A 0-A7 при перевірці внутрішнього ПЗП (РПЗП).	вхід/вихід
9	RST	Сигнал загального скидання. Вивід резервного живлення ОЗП від зовнішнього джерела (для 1816).	вхід
10-17	P3.0-P3.7	8-розрядний двонаправлений порт P3 з додатковими функціями:	вхід/вихід
	P3.0	Послідовні дані приймача - Rxd	вхід
	P3.1	Послідовні дані передавача - Txd	вихід
	P3.2	Вхід зовнішнього переривання 0- $\overline{INT} 0$	вхід
	P3.3	Вхід зовнішнього переривання 1- $\overline{INT} 1$	вхід
	P3.4	Вхід таймера/лічильника 0: - T0	вхід
	P3.5	Вхід таймера/лічильника 1: - T1	вхід
	P3.6	Вихід стробуючого сигналу АЛП при записі в зовнішню пам'ять даних: - $\overline{WR}$	вихід
	P3.7	Вихід стробуючого сигналу АЛП при читанні із зовнішньої пам'яті даних: - $\overline{RD}$	вихід
18 19	BQ2 BQ1	Виводи для підключення кварцового резонатора.	вихід вхід
20	0 V	Загальний вивід	
21-28	P2.0-P2.7	8-розрядний двонаправлений порт P2. Вихід адреси A 8-A15 у режимі роботи із зовнішньою пам'яттю. У режимі перевірки внутрішнього ПЗП виводи P2.0 - P2.6 використовуються як вхід адреси A8-A14. Вивід P2.7 - дозвіл читання ПЗП: - E	вхід/вихід
29	$\overline{PME}$	Дозвіл програмної пам'яті	вихід
30	ALE	Вихідний сигнал дозволу фіксації адреси. При програмуванні РПЗП сигнал: - PROG	вхід/вихід

Продовження таблиці 1.3

N виводу	Позначення	Призначення	Тип
31	DEM A	Блокування роботи із внутрішньою пам'яттю. При програмуванні РПЗП подається сигнал UPR	вхід/ вихід
32-39	P0.7-P0.0	8-розрядний двонаправлений порт P0. Шина адреси даних при роботі із зовнішньою пам'яттю. Вихід даних D7-D0 у режимі перевірки внутрішнього ПЗП (РПЗ).	вхід/ вихід
40	Ucc	Вивід живлення від джерела напруги +5 В.	

Структурна організація і система команд МК51 добре описана в [1] і наведена на рисунку 1.2. Відзначимо, що МК можна представити у вигляді наступних основних вузлів:

- арифметико-логічний пристрій (АЛП);
- пам'ять даних і програм;
- акумулятор;
- регістр слова стану (регістр ознак);
- 8 - бітний регістр - показчик стека;
- 16 - бітний регістр - показчик даних;
- блок таймерів - лічильників;
- послідовний порт;
- регістри спеціальних функцій;
- пристрою керування і синхронізації;
- порти введення - виводу.

МК51 працює від внутрішнього генератора, який має зовнішні виводи для підключення кварцового резонатора.

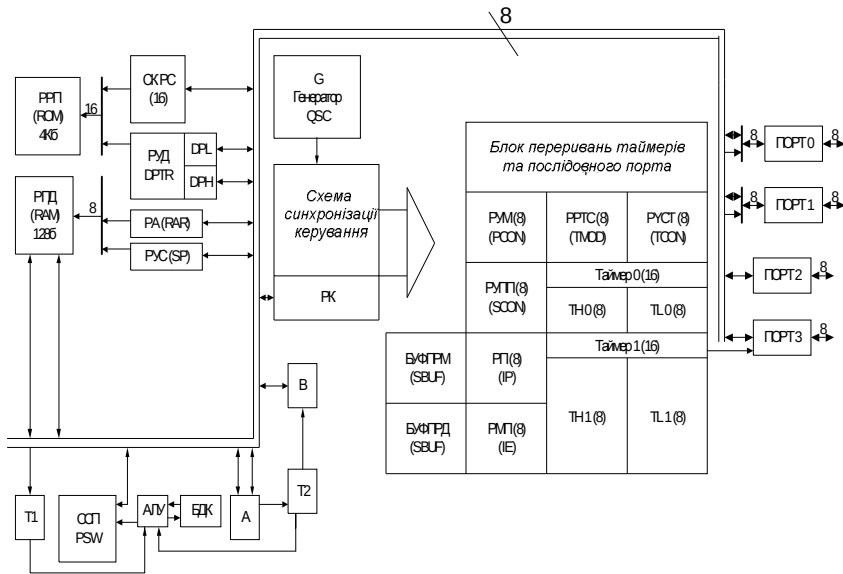


Рисунок 1.2 – Структурна схема мікроконтролера KP1816BE51 (МК51)

Порти (0, 1, 2, 3) МК51 служать для побайтного введення - виводу даних. Крім того, порти 0 і 2 служать для виводу адреси. Виводи порту 3 можуть бути використані для реалізації альтернативних функцій. Порт 0 є двонаправленим, порти 1, 2, 3 - квазідвонаправлені: кожна лінія порту може бути налаштована для введення або виводу даних. Навантажувальна здатність портів 1, 2, 3 - один ТТЛ - вхід, а порту 0 - два ТТЛ входу. РС - 16-розрядний програмний лічильник.

Акумулятор (А) служить для зберігання як операнда, так і результату операції. Але МК51 може виконувати ряд команд і без участі акумулятора. Виконання багатьох команд проводиться в арифметико-логічному пристрої АЛП, а ряд ознак операцій фіксується в регістрі слова стану програми (PSW)- РССП.

Арифметико-логічний пристрій (АЛП): 8-бітний може виконувати арифметичні операції додавання, вирахування, множення і ділення; логічні операції І, АБО, що виключає АБО, а також операції циклічного зрушення, скидання, інвертування і т.п. В АЛП є

програмно недоступні регістри T1 і T2, призначені для тимчасового зберігання операндів, схема десяткової корекції і схема формування ознак.

Восьмирозрядний показчик стека (SP) може адресувати будь-яку область пам'яті даних. Уміст його зменшується на одиницю в ході виконання команд PUSH і CALL і збільшується на одиницю при виконанні команд POP і RET. При скиданні в SP автоматично завантажується початковий код 07 H.

16-розрядний регістр-показчик даних (DPTR) служить для фіксації адреси при звертанні до зовнішньої пам'яті. Він може працювати як два незалежні 8-розрядних регістри DPH і DPL.

МК51 має два незалежні програмувальні 16-розрядних таймера/лічильника, представлених у вигляді двох регістрових пар: ТН0, ТЛ0 і ТН1, ТЛ1, буфер послідовного порту SBUF - два незалежні 8 - розрядних регістру: регістр приймача і регістр передавача, регістри спеціальних функцій з іменами IP, IE, TMOD, TCON, SCON, PCON, що дозволяють програмувати схему переривання, режими роботи таймерів-лічильників і прийомопередавача.

Пам'ять програм і пам'ять даних, розміщені на кристалі МК51, фізично і логічно розділені, мають різні механізми адресації, працюють під керуванням різних сигналів і виконують різні функції.

Пам'ять програм: постійний запам'ятовуючий пристрій (ПЗП) має ємність 4 Кбайта і призначений для зберігання: команд, констант, що управляють словами ініціалізації (при включенні), таблиць перекодування вхідних і вихідних змінних і т.п. **Регістр ПЗП** має 16-бітну шину адреси, через яку забезпечується доступ з лічильника команд або з **регістру-показчика даних (РПД)**. Останній виконує функції базового регістру при непрямих переходах по програмі або використовується в командах, що оперують із таблицями. Оперативний запам'ятовуючий пристрій (ОЗП) потрібен для зберігання змінних у процесі виконання прикладної програми, адресується одним байтом і має ємність 128 байт. Крім того, до адресного простору РПД примикають адреси **регістрів спеціальних функцій (РСФ)**, які перераховано в таблиці 1.4.

Таблиця 1.4 - Блок регістрів спеціальних функцій (РСФ)

Символ	Найменування	Адреса
* ACC	Акумулятор	0E0H
* B	Регістр-Розширник акумулятора	0F0H
* PSW	Слово стану програми	0D0H
SP	Регістр-Показчик стека	81H
DPTR	Регістр-Показчик даних (DPH)/(DPL)	83H/82H
* P0	Порт 0	80H
* P1	Порт 1	90H
* P2	Порт 2	0A0H
* P3	Порт 3	0B0H
* IP	Регістр пріоритетів	0B8H
* IE	Регістр маски переривань	0A8H
TMOD	Регістр режиму таймера/лічильника	89H
* TCON	Регістр керування/статус таймера	88H
TH0	Таймер 0 (старший байт)	8CH
TL0	Таймер 0 (молодший байт)	8AH
TH1	Таймер 1 (старший байт)	8DH
TL1	Таймер 1 (молодший байт)	8BH
* SCON	Регістр керування приймача	98H
SBUF	Буфер приймача	99H
PCON	Регістр керування потужністю	87H
Примітка. Регістри, які відзначені *, допускають адресацію окремих біт.		

1.4 Арифметико-логічний пристрій (АЛП). Регістр PSW

АЛП являє собою паралельний восьмирозрядний пристрій, що забезпечує виконання арифметичних і логічних операцій, а також операції логічного зрушення, скидання, установки і т.д..

АЛП складається з регістру акумулятора, регістру тимчасового зберігання, ПЗУ констант, суматора, додаткового регістру (регістру В), акумулятора, регістру стану програми.

Регістри акумулятора і тимчасового зберігання — восьмирозрядні регістри, призначені для приймання і зберігання операндів на час виконання операцій над ними. Програмно не доступні.

ПЗУ констант забезпечує створення коригувального коду при двійчно-десятковому представленні даних, коду маски при бітових операціях і коду констант.

Паралельний восьмирозрядний суматор являє собою схему комбінаційного типу з послідовним переносом, призначену для виконання арифметичних операцій додавання, вирахування і логічних операцій додавання, множення, нерівнозначності і тотожності.

Регістр В — восьмирозрядний регістр, використовується під час операцій множення і ділення. Для інших інструкцій він може розглядатися як додатковий надоперативний регістр.

Акумулятор являє собою восьмирозрядний регістр, призначений для приймання і зберігання результату, отриманого при виконанні арифметико-логічних операцій або операцій пересилання.

Регістр стану програми (PSW) призначений для зберігання інформації про стан АЛП при виконанні програми. Позначення розрядів регістру PSW і призначення розрядів наведені відповідно в таблиці 1.5.

Прапор переносу СУ може встановлюватися і скидатися як апаратними, так і програмними засобами. Прапор СУ може бути програмно прочитаний. Апаратними засобами прапор СУ встановлюється, якщо в старшому біті результату виникає перенос або позика. При виконанні операцій множення і розподілу прапор СУ скидається. Крім того, прапор СУ виконує функції "булевого акумулятора" у командах, що працюють із бітами.

Прапор додаткового переносу АС програмно доступний по запису ("0" і "1") і читанню.

Прапори F0, RS1, RS0 програмно доступні по запису ("0" і "1") і читанню. Прапор переповнення OV програмно доступний по запису ("0" і "1") і читанню. Встановлюється апаратно, якщо результат операції додавання/вирахування не укладається в семи бітах і старший (восьмий) біт результату не може інтерпретуватися як знаковий. При виконанні операції розподілу прапор OV апаратно скидається, а у випадку розподілу на нуль встановлюється. При множенні прапор OV апаратно встановлюється, якщо результат більше 255.

Прапор Р є доповненням вмісту акумулятора до парності. В 9-розрядному слові, що полягає з 8 розрядів акумулятора і біта Р, завжди втримується парне число одиничних бітів. У випадку, якщо в акумуляторі всі розряди встановлені в "0", прапор Р прийме нульове значення. Програмно доступний тільки по читанню.

У таблиці 1.5 приводиться перелік прапорів ФССП, даються їхні символічні імена і описуються умови їх формування. Особливості програмування регістрів наведено в [1,3].

Таблиця 1.5 - Формат слова стану програми (ФССП)

Символ	Позиція/№ біта	Ім'я і призначення			
CY	PSW.7	Прапор переносу. Встановлюється і скидається при виконанні арифметичних і логічних операцій			
AC	PSW.6	Прапор допоміжного переносу. Встановлюється і скидається при виконанні команд додавання і вирахування і сигналізує про перенос або позику в біті 3			
F0	PSW.5	Прапор 0. Може бути встановлений, скинутий або перевірений програмою як прапор, специфікуємий користувачем			
RS1 RS0	PSW.4 PSW.3	Вибір банку регістрів. Встановлюється і скидається програмою для вибору робочого банку регістрів			
OV	PSW.2	Прапор переповнення. Встановлюється і скидається при виконанні арифметичних операцій			
-	PSW.1	Не використовується			
P	PSW.0	Прапор паритету. Встановлюється і скидається в кожному циклі команди і фіксує непарне/парне число одиничних біт в акумуляторі			
Примітка		RS1	RS0	Банк	Границі адрес
		0	0	0	00H-07H
		0	1	1	08H-0FH
		1	0	2	10H-17H
		1	1	3	18H-1FH

1.5 Блок таймерів/лічильників. Регістри TMOD і TCON

Таймери/лічильники (Т/Л) призначені для підрахунку зовнішніх подій, для одержання програмно керованих тимчасових затримок і виконання часових функцій МК. До складу блоку Т/Л входять:

- два 16-розрядних регістри Т/Л 0 і Т/Л 1;
- восьмирозрядний регістр режимів Т/Л (TMOD);
- восьмирозрядний регістр керування (TCON);

- схема інкремента;
- схема фіксації $\overline{INT} 0$, $\overline{INT} 1$, T0, T1;
- схема керування прапорами;
- логіка керування T/C.

Два 16-розрядних регістри T/Л0 і T/Л1 виконують функцію зберігання вмісту рахунку. Кожний з них складається з пари восьмирозрядних регістрів, відповідно ТН0, ТЛ0 і ТН1, ТЛ1. Причому регістри ТН0, ТН1 — старші, а регістри ТЛ0, ТЛ1 — молодші 8 розрядів. Кожний з восьмирозрядних регістрів має свою адресу і може бути використаний як РОН, якщо Т/Л не використовуються (біт TR0 для Т/Л0 і біт TR1 для Т/Л1 у регістрі керування TCON рівні "0").

Код величини початкового рахунку заноситься в регістри Т/Лз програмно.

У процесі рахунку вміст регістрів Т/Л інкрементується. Ознакою закінчення рахунку, як правило, є переповнення регістру Т/З, тобто перехід його вмісту зі стану "усі одиниці" у стан "усі нулі". Усі регістри ТН0, ТН1, ТЛ0, ТЛ1 доступні по читанню, і при необхідності, контроль досягнення необхідної величини рахунку може виконуватися програмно.

Регістр режимів Т/Л (TMOD) призначений для приймання і зберігання коду, що визначає:

- один з 4-х можливих режимів роботи кожного Т/Л;
- роботу як таймери або лічильника;
- керування Т/Л від зовнішнього виводу.

Позначення розрядів регістру TMOD наведено в таблиці 1.6.

Таблиця 1.6 - Регістр **TMOD**

Біти	7	6	5	4	3	2	1	0
Позначен.	GATE 1	C/T1	ML1	M0.1	GATE0	C/T0	M1.0	M0.0

Призначення розрядів регістру TMOD наведено в таблиці 1.7.

Таблиця 1.7 - Регістр **TMOD**

Біти	Найменування	Призначення бітів			Примітка
1, 4, 5	M0-M1	Визначають один з 4-х режимів роботи, окремо для Т/Л1 і Т/Л0			Усі біти встановлюються програмно; біти 0-3 визначають режим роботи Т/Л0, біти 4-7 визначають режим роботи Т/Л1.
		M1	M0	Режим	
		0	0	0	
		0	1	1	
		1	0	2	
		1	1	3	
2, 6	C/T0 C/T1	Визначають роботу в якості: З/Т 0, З/Т 1= 0 - таймера C/Т 0, З/Т 1= 1 - лічильника			
3, 7	GATE	Дозволяє управляти таймером від зовнішнього виводу (0- $\overline{INT}$ для Т/Л0, $\overline{INT}$ 1- для Т/Л1). GATE=0- керування заборонене GATE= 1- керування дозволене			

При роботі в якості таймера вміст регістру T/Л інкрементується в кожному машинному циклі, тобто T/Л є лічильником машинних циклів МК. Оскільки машинний цикл складається з 12 періодів частоти синхронізації МК fbq , то частота рахунку в цьому випадку рівна $fbq/12$.

При роботі T/Л у якості лічильника зовнішніх подій вміст регістру T/Л інкрементується у відповідь на перехід з "1" в "0" сигнал АЛП на вході лічильника МК (вивід T0 для T/Л0 і вивід T1 для T/Л1). Входи лічильника апаратно перевіряються у фазі S5P2 кожного машинного циклу. Коли перевірки показують високий рівень на вході лічильника в одному машинному циклі і низький рівень в іншому машинному циклі, регістр T/Л інкрементується. Нове (інкрементоване) значення заноситься в регістр T/Л у фазі S3P1 машинного циклу, що безпосередньо впливає за тим, у якому був виявлений перехід з "1" в "0" на рахунковому вході МК. Так як для розпізнавання такого переходу потрібно два машинні цикли (24 періоду частоти синхронізації МК fbq), та максимальна частота рахунку T/Л у режимі лічильника рівна $fbq/24$.

Щоб рівень сигналу АЛП на рахунковому вході був гарантовано зафіксований, він повинен залишатися незмінним протягом як мінімум одного машинного циклу.

Регістр керування (TCON) призначений для приймання і зберігання коду керуючого слова. Позначення розрядів регістру TCON наведене в табл. 1.8. Призначення розрядів регістру TCON наведене в табл. 1.9.

Прапори переповнення TF0 і TF1 встановлюються апаратно при переповненні відповідних Т/Л (перехід Т/Л із стану "усі одиниці" у стан "усі нулі"). Якщо при цьому переривання від відповідного Т/Л дозволене, то установка прапора TF викличе переривання. Прапори TF0 і TF1 скидаються апаратно при передачі керування програми обробки відповідного переривання.

Прапори TF0 і TF1 програмно доступні і можуть бути встановлені/скинуті програмою. Використовуючи цей механізм, переривання по TF0 і TF1 можуть бути викликані (установка TF) і скасовані (скидання TF) програмою.

Прапори 1E0 і 1E1 встановлюються апаратно від зовнішніх переривань (відповідно входи МК $\overline{INT} 0$ і $\overline{INT} 1$) або програмно і ініціюють виклик програми обробки відповідного переривання. Скидання цих прапорів виконується апаратно при обслуговуванні переривання тільки в тому випадку, коли переривання було викликано по фронту сигналу АЛП. Якщо переривання було викликано рівнем сигналу АЛП на вході $\overline{INT} 0$ ($\overline{INT} 1$), то скидання прапора 1E повинна виконувати програма обслуговування переривання, впливаючи на джерело переривання для зняття їм запиту. Схема інкремента призначена:

- для збільшення на 1 у кожному машинному циклі вмісту регістрів Т/Л0, Т/Л1, для яких встановлений режим таймера і рахунок дозволений;

- для збільшення на 1 вмісту регістрів Т/Л0, Т/Л1, для яких установлений режим лічильника, рахунок дозволений і на відповідному вході МК (T0 для Т/Л0 і T1 для Т/Л1) зафіксований рахунковий імпульс.

Схема фіксації $\overline{INT} 0$, $\overline{INT} 1$, T0, T1 являє собою чотири тригери. У кожному машинному циклі в момент S5P2 у них запам'ятовується інформація з виводів МК INT0, $\overline{INT} 1$, T0, T1.

Схема керування прапорами виробляє і знімає прапори переповнення Т/С і прапори запитів зовнішніх переривань.

Логіка керування Т/Із синхронізує роботу регістрів Т/Л0 і Т/Л1 відповідно до запрограмованих режимів роботи і синхронізує роботу блоку Т/С з роботою МК .

Таблиця 1.8 - Регістр **TCON**

Біти	7	6	5	4	3	2	1	0
Позначення	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Таблиця 1.9 - Регістр **TCON**

Біти	Найменування	Призначення бітів	Примітка
6 4	TR1 TR0	Біти вимикання Т/З, окремо для Т/Л0 і Т/Л1. TR=0 - виключений, TR=1 - включений.	Біти встановлюються й скидаються програмно. Доступні по читанню.
7 5	TF1 TF0	Прапори переповнення Т/С.	Біти скидаються й встановлюються апаратно і програмно. Доступні по читанню.
2 0	IT1 IT0	Біти, що визначають вид переривання по входах $\overline{INT} 0, \overline{INT} 1$: IT=0 - переривання за рівнем (низькому) IT=1 - переривання по фронту (перехід з "1" в "0")	Біти встановлюються і скидаються програмно. Доступні по читанню.

Продовження таблиці 1.9

Біти	Найменування	Призначення бітів	Примітка
3 1	IE1 IE0	Прапори запиту зовнішніх переривань по входах $\overline{INT} 0$, $\overline{INT} 1$	Біти скидаються і встановлюються апаратно і програмно. Доступні по читанню.
			Біти 4,5 ставляться до Т/Л0; біти 6,7- до Т/Л1. Біти 0,1 визначають зовнішні переривання по входу $\overline{INT} 0$, біти 2,3 - по входу $\overline{INT} 1$.

Режим роботи кожного Т/Л визначається значенням бітів M0, M1 у регістрі TMOD. Т/Л0 і Т/Л1 мають чотири режими роботи. Режими роботи 0, 1, 2 однакові для обох Т/Л; Т/Л0 і Т/Л1 у цих режимах повністю незалежні друг від друга. Робота Т/Л0 і Т/Л1 у режимі 3 різна. При цьому установка режиму 3 у Т/Л0 впливає на режими роботи Т/Л1.

Установка бітів M0=0, M1=0 визначає режим роботи 0. Т/С у режимі 0 являє собою пристрій на основі 13-розрядного регістру.

13-розрядний регістр полягає для Т/Л0 з 8 розрядів регістру TH0 і 5 молодших розрядів регістру TL0, а для Т/Л1 - з 8 розрядів регістру TH1 і 5 молодших розрядів регістру TL1.

У цьому режимі функцію дільника на 32 виконують регістри TL0, TL1. Вони є програмно доступними, але треба пам'ятати, що значущими в режимі 0 є тільки п'ять молодших розрядів регістрів TL0, TL1. Логіка роботи в режимі 0 на прикладі Т/Л1 показана на рис. 1.3. Для Т/Л0 логіка роботи аналогічна. На рис. 1.3—1.6 OSC — джерело синхронізації МК (внутрішній або зовнішній). На виході OSC — частота fbq. Біт 3/Т регістру TMOD визначає роботу Т/Л або в якості таймера (3/Т=0), або в якості лічильника (3/Т=1). Рахунок починається при установці біта TR регістру TCON у стан "1". При необхідності керування рахунком ззовні біт GATE регістру TMOD встановлюється в стан "1". Тоді при TR=1 рахунок буде дозволений, якщо на вході $\overline{INT} 0$ (для Т/Л0) або $\overline{INT} 1$ (для Т/Л1) встановлений

стан "1" і буде заборонений, якщо встановлений стан "0". Установка біта TR0 для Т/Л0 і TR1 для Т/Л1 у стан "0" виключає Т/Л незалежно від стану інших бітів.

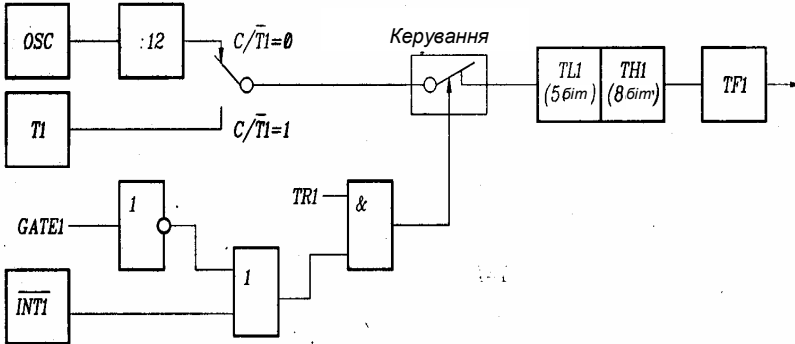


Рисунок 1.3 - Логіка роботи Т/Л1 у режимі 0

При переповненні Т/Л(перехід умісту регістру Т/Із зі стану "усі одиниці" у стан "усі нулі") встановлюється прапор TF0 для Т/Л0 або TF1 для Т/Л1 у регістрі TCON.

Установка бітів $M1=0$, $M0=1$ визначає режим роботи 1. Режим 1 аналогічний режиму 0. Відмінність полягає в тому, що установка режиму 1 перетворює Т/С у пристрій на основі 16-розрядного регістру. Для Т/Л0 регістр складається із програмно доступних пар TL0, TH0, для Т/Л1 із програмно доступних пар TL1, TH1. Логіка роботи в режимі 1 на прикладі Т/Л1 показана на рис. 1.4.

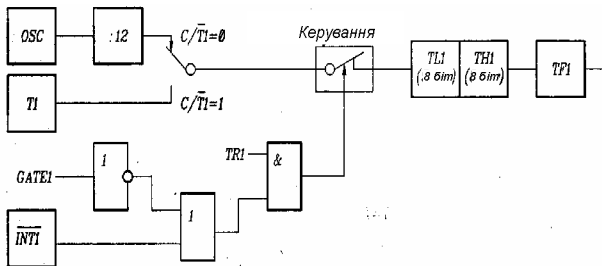


Рисунок 1.4 - Логіка роботи Т/Л1 у режимі 1

Установка бітів $M1=1$, $M0=0$ визначає режим 2. У цьому режимі Т/Л1 являє собою пристрій на основі восьмирозрядного регістру TL0 для Т/Л0 і TL1 для Т/Л1. При кожному переповненні TL0 крім установки в регістрі TCON прапора TF0 відбувається автоматично перезавантаження вмісту із TH0 в TL0. Відповідно для Т/Л1 при переповненні TL1 у регістрі TCON установлюється прапор TF1 і відбувається перезавантаження TL1 із TH1. Регістри TH0 і TH1 завантажуються програмно. Перезавантаження TL0 із TH0 і TL1 із TH1 не впливає на вміст регістрів TH0 і TH1. Логіка роботи Т/Л1 у режимі 2 показана на рис. 1.5. Логіка роботи Т/Л0 у режимі 2 аналогічна. Призначення бітів керування TR0, TR1, GATE0, GATE1, 3/T 0, 3/T 1 таке ж як режимі 0.

Установка бітів $M1=1$, $M0=1$ визначає режим 3. Т/Л1 у режимі 3 заблокований і просто зберігає свій рахунок (значення коду в регістрі Т/3). Ефект такий же, як при установці TR1=0.

Т/Л0 у режимі 3 являє собою два незалежні пристрої на основі восьмирозрядних регістрів TL0 і TH0. Пристрій на основі регістру TL0 може працювати в режимі таймера і у режимі лічильника. За ним зберігаються всі біти керування Т/Л0, воно реагує на впливи по входах T0, $\overline{INT} 0$. При переповненні TL0 встановлюється прапор TF0. Пристрій на основі регістру TH0 може працювати тільки в режимі таймера. Воно використовує біт включення TR1, при переповненні TH0 виставляє прапор TF1. Інших бітів керування пристрій на основі TH0 у цьому режимі не має. Логіка роботи Т/Л0 у режимі 3 показана на рис. 1.6.

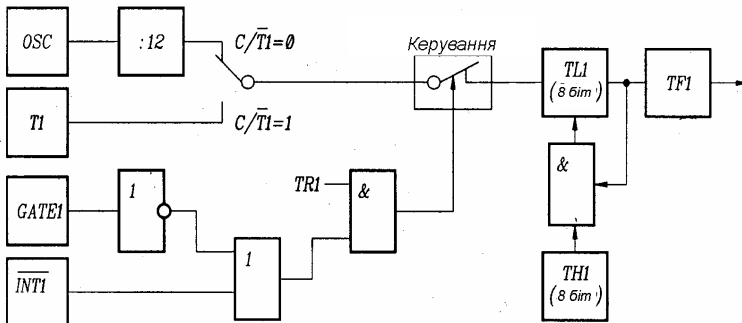


Рисунок 1.5 - Логіка роботи Т/Л1 у режимі 2

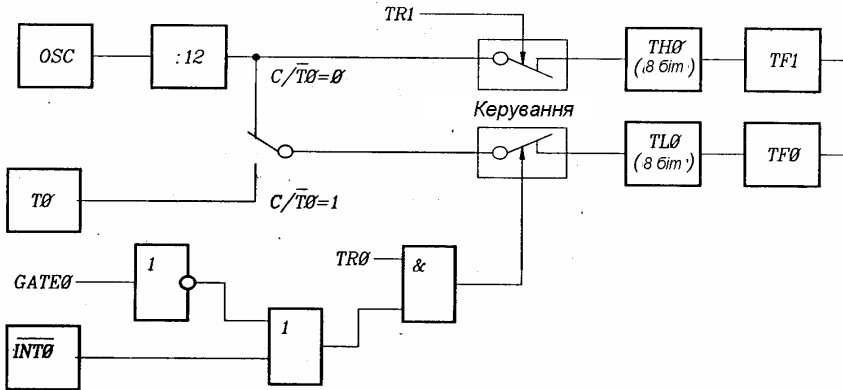


Рисунок 1.6 - Логіка роботи Т/Л0 у режимі 3

Установка Т/Л0 у режим 3 позбавляє Т/Л1 біта включення TR1. Тому Т/Л1 у режимах 0, 1, 2 при GATE1=0 завжди включений і при переповненні в режимах 0 і 1 Т/Л1 скидається, а в режимі 2 перезавантажується не встановлюючи прапор, якщо Т/Л0 перебуває в режимі 3. Керування від входів $\overline{INT}$ 1, T1, біти керування C/T1, GATE1 для Т/Л1 не залежать від режиму Т/Л0.

Т/Л1 апаратно пов'язаний із блоком синхронізації послідовного інтерфейсу (ПІ). При роботі в режимах 0, 1, 2 при переповненні Т/Л1 завжди виробляє тактовий імпульс ПІ. Тому 3-й режим Т/Л0 зручно застосовувати тоді, коли потрібна робота ПІ і двох таймерів або ПІ, таймера і лічильника.

Коли Т/Л0 переведено в режим 3, Т/Л1 можна виключити, перевіривши його також у режим 3, використовувати з послідовним портом для виробітку імпульсів тактировки або в будь-які інших додатках переривання, що не вимагають.

Додаткова інформація. Вимикання Т/Л за допомогою бітів TR0, TR1 (скидання цих бітів в "0") або за допомогою входів МК $\overline{INT}$ 0, $\overline{INT}$ 1 (установка на цих входах логічного "0" при GATE=1) не спотворює код, що перебуває в регістрі Т/Л. Т/Л можна виключити, через довільне, час знову включити і рахунок почнеться з тієї величини, яка була в регістрі Т/Л на момент вимикання (якщо, звичайно, після вимикання регістр Т/Л не перезаписувався)

Нове завантаження Т/Л відразу ж означає нову величину рахунку, а стара губиться. Якщо завантаження зроблене при включеному Т/З, рахунок продовжиться з нової величини. Черговість завантаження регістрів TL0, TH0, TL1, TH1 — довільна.

У всіх режимах, крім режиму 2, після переповнення Т/Л рахунок триває з величини 00H, якщо Т/Л не виключити за допомогою бітів TR0, TR1 або входів $\overline{INT}$ 0, $\overline{INT}$ 1.

1.6 Блок послідовного інтерфейсу і переривань. Регістри SCON, IP, IE

Блок послідовного інтерфейсу і переривань (ППП) призначений для організації введення-виводу послідовних потоків інформації і організації системи переривання програм.

До складу блоку PPP входять: буфер PPP, логіка керування PPP, регістр керування, буфер передавача, буфер приймача, приймач/передавач послідовного порту, регістр пріоритетів переривань, регістр дозволу переривань, логіка обробки прапорів переривань і схема виробітку вектора.

Буфер PPP забезпечує побайтовий обмін інформацією між внутрішньою магістраллю даних і шиною PPP.

Логіка керування PPP призначена для виробітку сигналів керування, що забезпечують чотири режими роботи послідовного інтерфейсу, і організації переривання програм.

Послідовний інтерфейс (послідовний порт) МК51 може працювати в наступних чотирьох режимах:

Режим 0. Інформація передається і ухвалюється через вхід приймача Rxd (вивід P3.0 МК). Через вихід передавача Txd (вивід P3.1 МК) видаються імпульси синхронізації, що стробують кожний переданий або прийнятий біт інформації. Формат посилки — 8 біт. Частота приймання і передачі — $fbq/12$, де fbq — тактова частота МК.

Режим 1. Інформація передається через вихід передавача Txd, а ухвалюється через вхід приймача Rxd. Формат посилки — 10 біт: старт-біт (нуль), вісім біт даних і стоп-біт (одиниця). Частота приймання і передачі задається Т/Л1.

Режим 2. Інформація передається через вихід передавача Txd, а ухвалюється через вхід приймача Rxd. Формат посилки — 11 біт: старт-біт (нуль), вісім біт даних, програмувальний дев'ятий біт і стоп-біт (одиниця). Переданий дев'ятий біт даних ухвалює значення біта

TB8 з регістру спеціальних функцій SCON. Біт TB8 у регістрі SCON може бути програмно встановлений в " ПРО" або в "1", або в нього, приміром, можна помістити значення біта Р з регістру PSW для підвищення вірогідності прийнятої інформації (контроль по паритету). При прийманні дев'ятий біт даних прийнятої посилки надходить у біт RB8 регістру SCON. Частота приймання і передачі в режимі 2 задається програмно і може бути рівна fbq/32 або fbq/64.

Режим 3. Режим 3 повністю ідентичний режиму 2 за винятком частоти приймання і передачі, яка в режимі 3 задається Т/Л1.

Докладно робота послідовного порту описано в розділі 2.3.3.

Регістр керування (SCON) призначений для приймання і зберігання коду восьмибітного слова, що управляє послідовним інтерфейсом. Позначення розрядів регістру SCON наведено в таблиці 1.10. Усі розряди регістру SCON програмно доступні по запису ("0" і "1") і читанню.

Таблиця 1.10 - Регістр керування послідовним інтерфейсом **SCON**

Біти	7	6	5	4	3	2	1	0
Позначення	SM0	SMI	SM2	REN	TB8	RB8	T1	RI

Таблиця 1.11 - Регістр керування послідовним інтерфейсом **SCON**

SM0	SMI	Режим	Найменування	Швидкість передачі
0	0	0	Регістр зсуву	fbq/12
0	1	1	8-бітовий універсальний асинхронний приймач/передавач (УАПП)	змінна, задається Т/Л1
1	0	2	9-бітовий (УАПП)	fbq/64 або fbq /32
1	1	3	9-бітовий (УАПП)	змінна, задається Т/Л1

Розряди SM0, SMI визначають режим роботи ППП, як зазначено в табл. 1.11. Інші біти регістру мають наступне призначення:

SM2 — дозвіл багатопроцесорної роботи. У режимах 2 і 3 при SM2=1 прапор RI не активізується, якщо дев'ятий прийнятий біт даних рівний "0". У режимі 1 при SM2=1 прапор RI не активізується, якщо не прийнятий стоп-біт, рівний "1". У режимі 0 біт SM2 повинен бути встановлений в "0".

REN — дозвіл приймання послідовних даних. Встановлюється і скидається програмним забезпеченням відповідно для дозволу і заборони приймання.

TB8 — дев'ятий біт переданих даних у режимах 2 і 3. Встановлюється і скидається програмним забезпеченням.

RB8 — дев'ятий біт прийнятих даних у режимах 2 і 3. У режимі 1, якщо SM2=0, RB8 є прийнятим стоп-бітом. У режимі 0 біт RB8 не використовується.

TI — прапор переривання передавача. Встановлюється апаратно наприкінці часу видачі 8-го біта в режимі 0 або на початку стоп-біта в інших режимах. Скидається програмним забезпеченням.

RI — прапор переривання приймача. Встановлюється апаратно наприкінці часу приймання 8-го біта в режимі 0 або через половину інтервал біту в режимах 1, 2, 3 при SM2=0. При SM2=1 див. опис для біта SM2.

Буфер передавача призначений для приймання із шини ППП паралельної інформації і видачі її у вигляді послідовного потоку символів на передавач послідовного порту.

Буфер приймача призначений для приймання даних у вигляді послідовного потоку символів з послідовного порту, перетворення їх у паралельний вид, зберігання і видачі в паралельному виді на внутрішню шину ППП.

Буфер приймача і буфер передавача при програмному доступі мають однакове ім'я (SBUF) і адреса (99H). Якщо команда використовує SBUF як регістр джерела, то обіг відбувається до буфера приймача. Якщо команда використовує SBUF як регістр призначення, то обіг відбувається до буфера передавача.

У всіх чотирьох режимах роботи послідовного порту передача ініціюється будь-якою командою, яка використовує SBUF як регістр призначення. Приймання в режимі 0 ініціюється умовою RI=0 і REN=1. В інших режимах приймання ініціюється приходом старт-біта, якщо REN=1.

Приймач/передавач послідовного порту призначений для приймання послідовного потоку символів із входу послідовного порту, виділення даних і видачі їх у буфер приймача, а також для приймання послідовних даних з буфера передавача, перетворення їх у послідовний потік символів і видачі його на вихід послідовного порту.

Регістр пріоритетів переривань (IP) призначений для установки рівня пріоритету переривання для кожного з п'яти джерел переривань. Позначення розрядів регістру IP показано в таблиці 1.12, а їх призначення зазначене нижче.

PX0 — установка рівня пріоритету переривання від зовнішнього джерела $\overline{INT}$ 0.

PT0 — установка рівня пріоритету переривання від T/Л0.

PX1 — установка рівня пріоритету переривання від зовнішнього джерела $\overline{INT}$ 1.

PT1 — установка рівня пріоритету переривання від T/Л1.

PS — установка рівня пріоритету переривання від послідовного порту.

X — резервний розряд.

Таблиця 1.12 - Регістр пріоритетів переривань IP

Біти	7	6	5	4	3	2	1	0
Позначення	X	X	X	PS	PT1	PX1	PT0	PX0

Наявність у розряді IP "1" встановлює для відповідного джерела високий рівень пріоритету, а наявність у розряді IP "0" — низький рівень пріоритету. При читанні резервних розрядів відповідні лінії магістралі даних не визначені. Користувач не повинен записувати "1" у резервні розряди, тому що вони зарезервовані під подальше розширення сімейства МК51.

Регістр дозволу переривань (IE) призначений для дозволу або заборони переривань від відповідних джерел. Позначення розрядів регістру IE показане, у таблиці 1.13, а їх призначення зазначене нижче.

X — резервний розряд.

EA — керування всіма джерелами переривань одночасно. Якщо EA=0, то переривання заборонені. Якщо EA=1, то переривання

можуть бути дозволені індивідуальними дозволами EX0, ET0, EX1, ET1, ES.

ES — керування перериванням від послідовного порту. ES=1 — дозвіл. ES=0 — заборона.

ET1 — керування перериванням від Т/Л1. ET1=1 — дозвіл. ET1=0 — заборона.

EX1 — керування перериванням від зовнішнього джерела $\overline{INT}$ 1. EX1=1 — дозвіл. EX1=0 — заборона.

ET0 — керування перериванням від Т/Л0. ET0=1 — дозвіл. ET0=0 — заборона.

EX0 — керування перериванням від зовнішнього джерела $\overline{INT}$ 0. EX0=1 — дозвіл. EX0=0 — заборона.

При читанні резервних розрядів відповідні лінії магістралі не визначені. Користувач не повинен записувати "1" у резервні розряди, тому що вони зарезервовані під подальше розширення сімейства МК51.

Таблиця 1.13 - Регістр дозволу переривань IE

Біти	7	6	5	4	3	2	1	0
Позначення	EA	X	X	ES	ET1	EX1	ET0	EX0

Логіка обробки прапорів переривань здійснює пріоритетний вибір запиту переривання, скидання його прапора і ініціює виробіток апаратно реалізованої команди переходу на підпрограму обслуговування переривання.

Схема виробітку вектора переривання виробляє двохбайтові адреси підпрограм обслуговування переривання залежно від джерела переривань, які наведено в таблиці 1.14. Докладно система переривань описана в розділі 2.3.4.

Таблиця 1.14 - Двохбайтові адреси підпрограм обслуговування переривання

Джерело переривання	Вектор переривання
Зовнішнє переривання $\overline{INT}$ 0	0003H
Таймер/лічильник Т/Л0	000BH
Зовнішнє переривання $\overline{INT}$ 1	0013H

Таймер/лічильник Т/Л1	001ВН
Послідовний порт	0023Н

1.7 Порти

Порти P0, P1, P2, P3 є двонаправленими портами введення-виводу і призначені для забезпечення обміну інформацією МК із зовнішніми пристроями, утворюючи 32 лінії вводу-виводу. Кожний з портів містить фіксатор-засувку, який являє собою восьмиразрядний регістр, що має байтову і бітову адресацію для установки (скидання) розрядів за допомогою програмного забезпечення.

Фізичні адреси фіксаторів P0, P1, P2, P3 становлять для:

- P0 — 80Н, при бітовій адресації 80 Н-Н-87Н;
- P1 — 90Н, при бітовій адресації 90 Н-Н-97Н;
- P2 — А0Н, при бітовій адресації А0 Н-А7Н;
- P3 — В0Н, при бітовій адресації В0 Н-В7Н.

Крім роботи як звичайних портів введення/виводу лінії портів P0-P3 можуть виконувати ряд додаткових функцій, описаних нижче.

Через порт P0: виводиться молодший байт адреси А0-А7 при роботі із зовнішньою пам'яттю програм і зовнішньою пам'яттю даних. Видається з МК і ухвалюється в МК байт даних при роботі із зовнішньою пам'яттю (при цьому обмін байтом даних і вивід молодшого байта адреси зовнішньої пам'яті мультиплексовані в часі);

Задаються дані при програмуванні внутрішнього ППЗУ і читається вміст внутрішньої пам'яті програм.

Через порт P1: задається молодший байт адреси при програмуванні внутрішнього ППЗУ і при читанні внутрішньої пам'яті програм.

Через порт P2: виводиться старший байт адреси А8-А15 при роботі із зовнішньою пам'яттю програм і зовнішньою пам'яттю даних (для зовнішньої пам'яті даних — тільки при використанні команд MOVX A, @DPTR і MOVX @DPTR, A, які виробляють 16-у адресу);

Задається старший байт (розряди А8-А14) адреси при програмуванні внутрішнього ППЗП і при читанні внутрішньої пам'яті програм.

Кожна лінія порту P3 має індивідуальну альтернативну функцію:

P3.0 - Rxd, вхід послідовного порту, призначений для введення послідовних даних у приймач послідовного порту;

P3.1 — Txd, вихід послідовного порту, призначений для виводу послідовних даних з передавача послідовного порту;

P3.2 — $\overline{INT}$ 0, використовується як вхід 0 зовнішнього запиту переривання;

P3.3 — $\overline{INT}$ 1, використовується як вхід 1 зовнішнього запиту переривання;

P3.4 — T0, використовується як вхід лічильника зовнішніх подій T/Л0;

P3.5 — T1, використовується як вхід лічильника зовнішніх подій T/Л1;

P3.6 — $\overline{WR}$, строб запису в зовнішню пам'ять даних, вихідний сигнал, що супроводжує вивід даних через порт P0 при використанні команд MOVX @Ri, A і MOVX @DPTR, A.

P3.7 — $\overline{RD}$, строб читання із зовнішньої пам'яті даних, вихідний сигнал, що супроводжує введення даних через порт P0 при використанні команд MOVX A, @Ri і MOVX A, @DPTR.

Альтернативна функція кожної з ліній порту P3 реалізується тільки в тому випадку, якщо у відповідному цій лінії розряді фіксатора-засувки втримується "1". А якщо ні, то на лінії порту P3 буде присутній "0".

1.8 Пам'ять даних

Пам'ять даних призначена для приймання, зберігання і видачі інформації, використовуваної в процесі виконання програми. Пам'ять даних, розташована на кристалі МК, складається з регістру адреси ОЗУ, дешифратора, ОЗП, покажчика стека.

Регістр адреси ОЗП призначений для приймання і зберігання адреси обираної за допомогою дешифратора комірки пам'яті, яка може містити як біт, так і байт інформації.

ОЗУ являє собою 128 восьмиразрядних регістрів, призначених для приймання, зберігання і видачі різної інформації.

Покажчик стека являє собою восьмиразрядний регістр, призначений для приймання і зберігання адреси гнізда стека, до якої був останній обіг. При виконанні команд LCALL, ACALL вміст

покажчика стека збільшується на 2. При виконанні команд RET, RETI вміст покажчика стека зменшується на 2. При виконанні команди PUSH direct вміст покажчика стека збільшується на 1. При виконанні команди POP direct вміст покажчика стека зменшується на 1. Після скидання в покажчику стека встановлюється адресу 07H, що відповідає початку стека з адресою 08H.

В МК передбачена можливість розширення пам'яті даних шляхом підключення зовнішніх пристроїв ємністю до 64 Кбайт. При цьому звертання до зовнішньої пам'яті даних можливо тільки за допомогою команд MOVX.

Команди MOVX @Ri, A і MOVX A, @Ri формують восьмирозрядну адресу, видавану через порт P0. Команди MOVX @DPTR, A и MOVX @A, DPTR формують 16-розрядну адресу, молодший байт якій видається через порт P0, а старший — через порт P2.

Байт адреси, видаваний через порт P0, повинен бути зафіксований у зовнішньому регістрі по спаду сигнал АЛП ALE, тому що надалі лінії порту P0 використовуються як шина даних, через яку байт даних ухвалюється з пам'яті при читанні або видається на згядку даних при записі. При цьому читання стробується сигналом МК $\overline{RD}$, а запис — сигналом МК $\overline{WR}$. При роботі із внутрішньою пам'яттю даних, сигнали $\overline{RD}$ і $\overline{WR}$ не формуються.

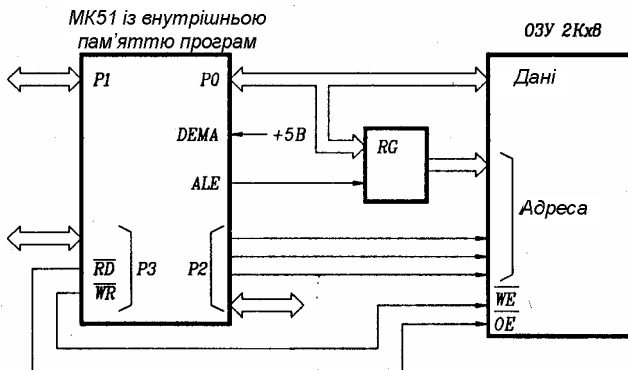


Рисунок 1.7 - Сторінкова організація зовнішньої пам'яті даних

На рис. 1.7 показана сторінкова організація зовнішньої пам'яті даних. Наведена схема дозволяє працювати з пам'яттю даних ємністю 2 Кбайт, використовуючи команди типу MOVX @Ri. Порт P0 при цьому працює як мультиплексована шина адресу/даних, а три лінії порту P2 адресують сторінки зовнішнього ОЗУ. Інші 5 ліній порту P2 можуть використовуватися в якості ліній введення/виводу. Докладно організація пам'яті даних описано в розділі 2.3.5.

На рис. 1.8 відповідно наведені діаграми циклів читання і запису при роботі МК із зовнішньою пам'яттю даних (Z — високоімпедансний стан).

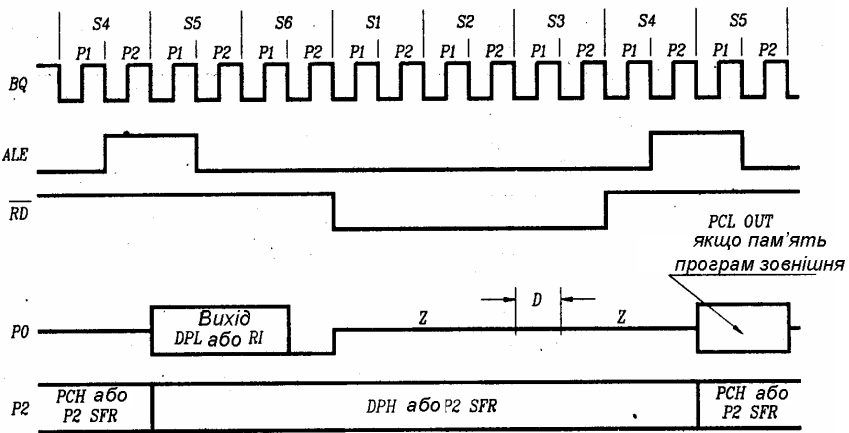


Рисунок 1.8 - Цикл читання із зовнішньої пам'яті даних

PCL OUT — видача молодшого байта лічильника команд PC;

PCH — старший байт лічильника команд PC;

DPL, DPH — відповідно молодший і старший байти регістру показчика даних DPTR, який використовується як регістр непрямої адреси в командах MOVX A, @DPTR і MOVX @DPTR, A;

P2 SFR — засувки порту P2;

RI — регістри P0 і R1, які використовуються в якості регістрів непрямої адреси в командах MOVX A, @RI і MOVX @P1, A;

D — період, протягом якого дані з P0 уводяться в МК.

1.9 Пам'ять програм

Пам'ять програм призначена для зберігання програм і має окреме від пам'яті даних адресний простір обсягом до 64 Кбайт, причому, для мікросхем KP1816BE51, KM1816BE751 і для KP1830BE51 частина пам'яті програм з адресами 0000 H-H—0FFFFH розташована на кристалі МК. Пам'ять програм, розташована на кристалі, складається з 12-розрядного дешифратора і ПЗП ємністю 4 до *8 біт для мікросхем KP1816BE51, KP1830BE51 або ППЗП з ультрафіолетовим стиранням ємністю 4Kx8 біт для KM1816BE751.

Запис програм у ПЗП відбувається під час виготовлення кристалів.

Якщо на вивід МК DEMA подана напруга живлення UCC, то звертання до зовнішньої пам'яті програм відбувається автоматично при виробітку лічильником команд адреси, що перевищує 0FFFFH. Якщо адреса перебуває в межах 0000 H-H-0FFFFH, обіг відбувається до пам'яті програм, розташованої на кристалі (внутрішньої пам'яті програм).

Якщо на вивід МК DEMA поданий "0", внутрішня пам'ять програм відключається і починаючи з адреси 0000H усі обіги виконуються до зовнішньої пам'яті програм.

PCL OUT — видача молодшого байта лічильника команд PC;

PCN — старший байт лічильника команд PC;

DPL, DPH — відповідно молодший і старший байти регістру показчика даних DPTR, який використовується як регістр непрямої адреси в командах MOVX A, @DPTR і MOVX @DPTR.A;

P2 SFR — засувки порту P2;

Ri — регістри R0 і R1, які використовуються в якості регістрів непрямої адреси в командах MOVX A, @Ri і MOVX @Ri, A.

Якщо МК не має внутрішньої пам'яті програм, її вивід DEMA повинен бути підключений до шини OB.

Читання із зовнішньої пам'яті програм стробується сигналом МК $\overline{PME}$. При роботі із внутрішньою пам'яттю програм сигнал не формується. МК не мають інструкцій і апаратних засобів для програмного запису на згадку програм.

При звертаннях до зовнішньої пам'яті програм завжди формується 16-розрядна адреса, молодший байт якого видається через порт P0, а старший — через порт P2. При цьому байт адреси,

видаваний через порт P0, повинен бути зафіксований у зовнішньому регістрі по спаду сигналу ALE, тому що надалі лінії порту P0 використовуються в якості шини даних, по якій байт із зовнішньої пам'яті програм уводиться в МК.

На рис. 1.9 показана функціональна схема включення МК МК51 із зовнішнім ППЗУ програм. Порт P0 працює як мультиплексована шина адресу/даних: видає молодший байт лічильника команд, а потім переходить у високоімпедансний стан і очікує приходу байта із ППЗУ програм. Коли молодший байт адреси перебуває на виходах порту P0, сигнал ALE записує його в адресному регістрі RG. Старший байт адреси перебуває на виходах порту P2 протягом усього часу звертання до ППЗУ. Сигнал дозволяє вибірку байта із ППЗУ, після чого обраний байт надходить на порт P0 МК51 і вводиться в МК.

Докладно організація пам'яті програм описано в розділі 2.3.5.

На рисунках 1.10 і 1.11 наведені діаграми, що показують формування відповідних сигналів при роботі МК із зовнішньою пам'яттю програм. Як видно з діаграм, при роботі із зовнішньою пам'яттю програм сигнал $\overline{PME}$ формується двічі в кожному машинному циклі незалежно від кількості байт у команді. Якщо другий обраний байт у поточній команді не використовується, він ігнорується МК. Надалі при переході до виконання наступної команди цей байт буде введений удруге.

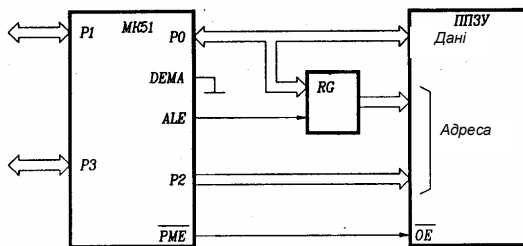
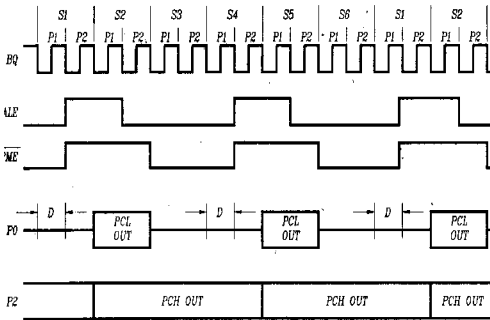


Рисунок 1.9 - Схема включення МК51 із зовнішнім ППЗУ програм



PCL OUT - видача молодшого байту лічильника команд PC
 PCH OUT - видача старшого байту лічильника команд PC
 D - періоди, під час яких дані с P0 вводяться в ОМЭВМ

Рисунок 1.10 - Робота із зовнішньою пам'яттю програм

Якщо виконується команда MOVX (рис. 1.11, діаграма В), два сигнали $\overline{PME}$ не формуються, тому що порт P0 звільняється для адресації і обміну даними із зовнішньою пам'яттю даних.

Коли МК працює із внутрішньою пам'яттю програм, $\overline{PME}$ не формується і адреса на портах P0 і P2 не видається. Проте, сигнал ALE буде формуватися двічі в кожному машинному циклі завжди за винятком випадку команди MOVX (у цьому випадку один сигнал ALE пропускається). Таким чином, якщо не використовуються команди MOVX, сигнал ALE може бути задіяний у якості вихідного синхросигналу.

На рисунку 11.1 подані цикли роботи із зовнішньою пам'яттю програм:

(А) - без виконання команди MOVX;

(В) – з виконанням команди MOVX;

PCL OUT — видача молодшого байту лічильника команд PC;

PCH OUT — видача старшого байту лічильника команд PC;

DPH — старший байт регістру покажчика даних DPTR, який використовується у якості регістру непрямого адресу в командах MOVX A, @DPTR і MOVX @DPTR, A;

P2 SFR — засувки порту P2;

INS IN— ввод байту інструкції з пам'яті програм;

ADDR OUT — видача молодшого байту адреси зовнішньої пам'яті даних з регістрів R0, R1 або регістру DPL;

Задній фронт ALE стробує адресу в порті P0. В цей час адреса гарантовано встановлена.

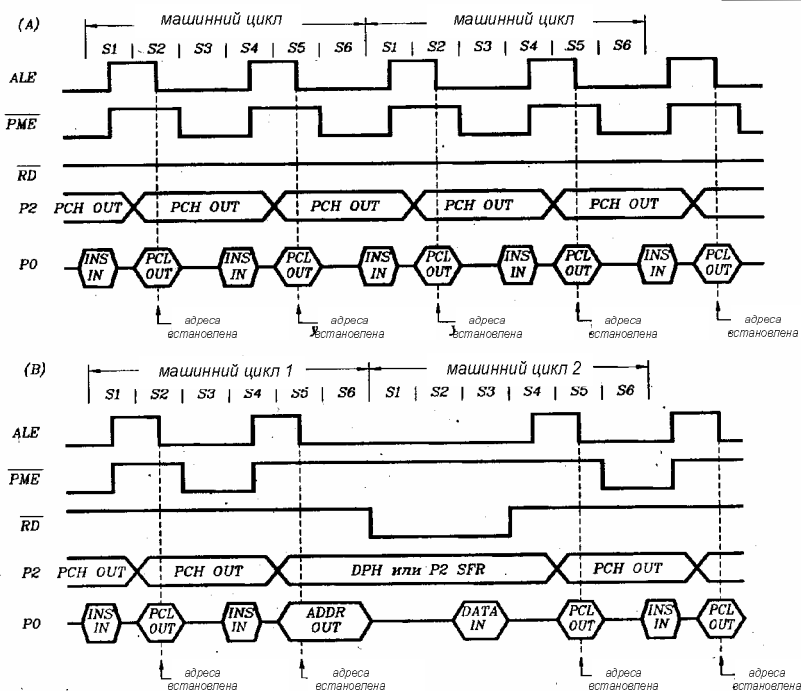


Рисунок 1.11 - Цикли роботи із зовнішньою пам'яттю програм

1.10 Блок керування. Синхронізація МК. Регістр PCON. Режими зменшеного енергоспоживання

Блок керування призначений для виробітку синхронізуючих і керуючих сигналів, що забезпечують координацію спільної роботи блоків МК у всіх припустимих режимах його роботи.

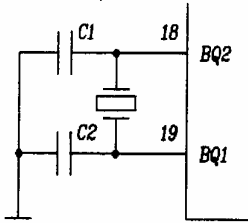
До складу блоку керування входять: пристрій виробітку тимчасових інтервалів, логіка введення-виводу, реєстр команд, реєстр керування споживанням, дешифратор команд, ПЛІМ і логіка керування ЕОМ.

Пристрій виробітку часових інтервалів призначений для формування і видачі внутрішніх синхросигналів фаз, тактів і циклів. Кількість машинних циклів визначає тривалість виконання команд.

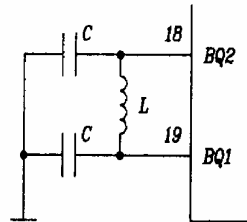
Практично всі команди МК виконуються за один або два машинні цикли, крім команд множення MUL A, Y і розподілу DIV A, B, тривалість виконання яких становить чотири машинні цикли. Машинний цикл має фіксовану тривалість і містить шість станів S1-S6, кожне з яких по тривалості відповідає такту, і, у свою чергу, складається із двох часових інтервалів, обумовлених фазами P1 і P2. Тривалість фази дорівнює періоду проходження зовнішнього сигналу BQ, що є первинним сигналом синхронізації МК. Сигнал BQ виробляється або вбудованим тактовим генератором МК при підключенні до її виводів 18 (BQ2) і 19 (BQ1) кварцового резонатора або LC-ланцюжка, або зовнішнім джерелом тактових сигналів.

Схема підключення кварцового резонатора і LC-Ланцюжка до виводів МК BQ2 і BQ1 показана на рис. 1.12. Схема підключення зовнішнього джерела тактових сигналів показана на рис. 1.13. Джерело тактових сигналів повинен забезпечувати наступні характеристики зовнішнього синхросигналу МК: — тривалість низького рівня сигнал АЛП — не менше 20 нс; — тривалість високого рівня сигнал АЛП — не менше 20 нс; — часи фронтів наростання і спаду сигнал АЛП — не більше 20 нс. Як видно з рис. 1.13, зовнішній синхросигнал для п-МОН МК (серія 1816) подається на вихід 18 (BQ2), а для КМОН МК (серія 1830) — на вихід 19 (BQ1). При цьому необхідно забезпечувати необхідні рівні напруги синхросигналу.

На рис. 1.14 показані в загальному виді внутрішні генератори п-моп і КМОН МК сімейства МК51. BQ1 і BQ2 є відповідно входом і виходом підсилювача, що інвертує, який може бути включений у режим генератора при підключенні до виводів BQ1 і BQ2 резонатора або LC-ланцюжка рис. 1.12.



$C1, C2$ – ємність $30\text{пФ} \pm 10\text{пФ}$



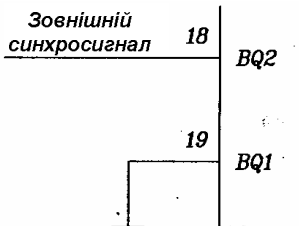
$$f = \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C+3C_{pp}}{2}$$

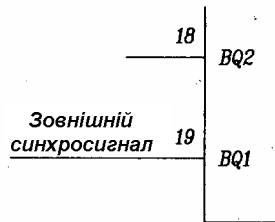
$C_{pp} \approx 10\text{пФ}$ – ємність вивода

Рисунок 1.12 - Підключення кварцового резонатора і LC-ланцюжка

Показаний на рис. 1.14 сигнал PD задається установкою однойменного біта в регістрі PCON для перекладу МК у режим мікроспоживання.



Для ОМЭВМ серії 1816



Вивод 18 (BQ2) залишити не підключеним

Для ОМЭВМ серії 1830

Рисунок 1.13 - Підключення зовнішнього джерела тактових сигналів

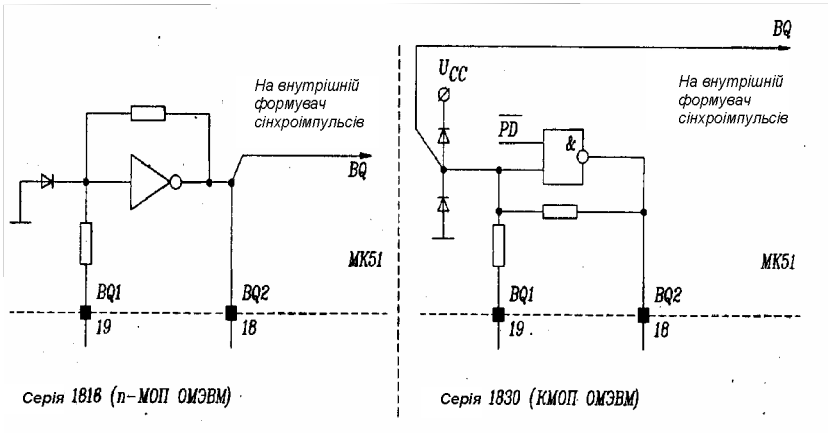


Рисунок 1.14 - Внутрішні генератори МК51

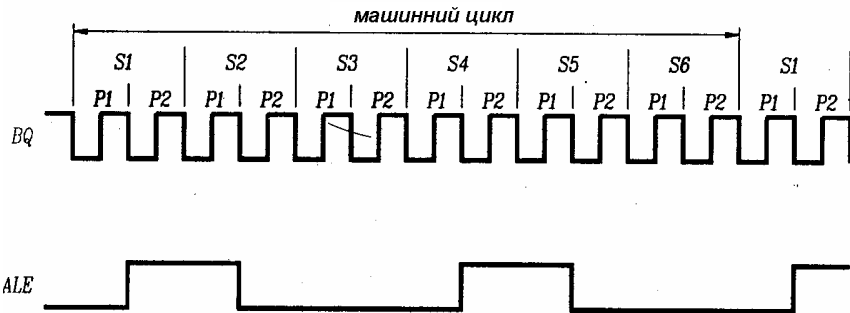


Рисунок 1.15 - Діаграма формування машинних циклів МК51

Рисунок 1.15 ілюструє формування машинних циклів МК. Усі машинні цикли МК однакові, складаються з 12 періодів сигналу BQ, починаються фазою S1P1 і закінчуються фазою S6P2. Двічі за один машинний цикл формується сигнал ALE. На рис. 1.17 показана діаграма зовнішнього синхросигналу МК.

Логіка введення-виводу призначена для приймання і видачі сигналів, що забезпечують обмін інформацією МК із зовнішніми пристроями через порти введення-виводу P0-P3.

Регістр команд призначений для запису і зберігання 8-и розрядного коду операції виконуваної команди, який за допомогою дешифратора команд перетворюється в 24-х розрядний код для ПЛМ, за допомогою якої виробляється набір мікрооперацій відповідно до мікропрограми виконання команди. Регістр команд програмно не доступний.

Конструкція регістру керування споживанням (PCON) визначається технологією виготовлення МК: п-МОН або КМОН.

Для варіанта виготовлення за технологією п-моп (серія 1816) регістр PCON має всього 1 біт, керуючий швидкістю передачі послідовного порту SMOD.

Для варіанта виготовлення за технологією КМОН (серія 1830) позначення розрядів регістру PCON наведено в таблиці 1.15, а призначення розрядів у таблиці 1.16.

Для п-моп і КМОН МК розташування і призначення розряду SMOD ідентичні.

Усі біти регістру PCON програмно доступні по запису ("0" і "1") і читанню.

Функції біта SMOD докладно розглянуті при описі роботи послідовного порту.

Таблиця 1.15 - Регістр **PCON** серії 1830 (КМОП)

Біти	7	6	5	4	3	2	1	0
Позначення	SMOD	-	-	-	GF1	GF0	PD	IDL

Таблиця 1.16 - Регістр **PCON** серії 1830 (КМОП)

Біти	Наиенов.	Призначення бітів	Примітка
7	SMOD	Біт подвоєння швидкості передачі: при установці в "1"- швидкість передачі подвоюється	При роботі послідовного порту
6	-	Резервний	
5	-	Резервний	
4	-	Резервний	
3	GF1	Прапор загального призначення	
2	GF0	Прапор загального призначення	

1	PD	Біт включення режиму мікроспоживання "1" - режим мікроспоживання	Якщо в PD і IDL одночасно записана "1", перевага має PD
0	IDL	Біт холостого ходу "1" - режим холостого ходу	

Біти PCON з номерами 4—6 зарезервовані для подальшого розширення сімейства МК51. При читанні значення цих розрядів не визначене. Програміст не повинен записувати "1" у ці біти, тому що вони можуть використовуватися в майбутніх розробках МК сімейства МК51 для завдання нових функцій. У цьому випадку пасивне значення бітів 4—6 буде "0", а активне — "1". Біти GF1 і GF0 користувач може задіяти за своїм розсудом. В МК сімейства МК51, виконаних по КМОН технології, є два режими зменшеного енергоспоживання: режим холостого ходу і режим мікроспоживання. Джерелом живлення в цих режимах є вивід Ucc.

Режими зменшеного споживання для КМОП МК ініціюються установкою бітів PD і IDL у регістрі PCON. Вплив цих бітів на апаратуру МК ілюстровано на рис. 1.16.

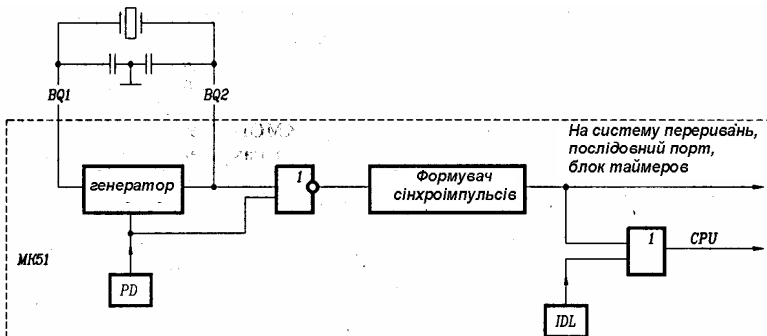


Рисунок 1.16 - Дія бітів PD и IDL регістру PCON

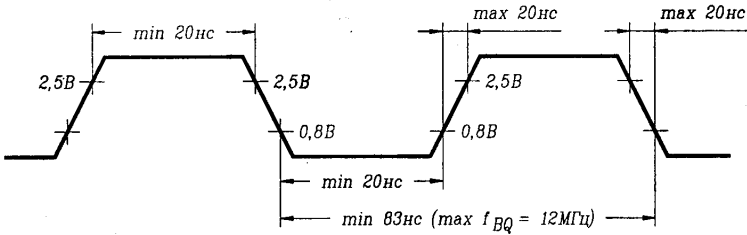


Рисунок 1.17 - Діаграма зовнішнього синхросигналу МК

Режим холостого ходу: інструкція, яка встановлює $PCON.0=1$ (IDL), є останньою інструкцією, виконуваною перед переходом у режим холостого ходу. У цьому режимі блокуються функціональні вузли центрального процесора (CPU), що і зменшує енергоспоживання. Зберігаються стани покажчика стека, програмного лічильника, PSW, акумулятора і усіх інших регістрів, а також внутрішнього ОЗП даних.

Для закінчення режиму холостого ходу є два способи. Активізація будь-якого дозволеного переривання автоматично приведе до установки $PCON.0=0$, закінчуючи режим холостого ходу. Після виконання команди RETI (вихід з підпрограми обслуговування переривання) буде виконана команда, яка іде за командою, яка переведе МК у режим холостого ходу.

Біти GF0 і GF1 зручно використовувати для індикації режиму, у якому була викликана програма обробки переривання: відбулося це при нормальній роботі МК або в режимі холостого ходу. Приміром, команда, що викликає режим холостого ходу, може також установлювати один або кілька прапорів (GF0, GF1 або яких-небудь інших). Програма обробки переривання, перевіряючи ці прапори, може визначити передісторію свого виклику.

Іншим способом закінчення режиму холостого ходу є апаратне скидання по входу RST тривалістю не менш двох машинних циклів.

Активний сигнал скидання на виводі RST асинхронно скидає біт IDL ($PCON.0$). Оскільки тактовий генератор працює, МК відразу після скидання IDL починає виконувати програму з команди, що впливає за командою, що викликала режим холостого ходу. Між скиданням біта IDL і моментом, коли ввімкнеться внутрішній алгоритм скидання, може пройти до двох машинних циклів виконання програми.

Внутрішні апаратні засоби МК блокують доступ до внутрішньої пам'яті даних протягом зазначеного часу, але не блокують доступ до портів. Якщо при цьому зміна інформації на портах небажана, то необхідно стежити, щоб за командою, яка встановлює біт IDL, не впливала безпосередньо команда, що записує інформацію в порт або в зовнішню пам'ять даних.

Режим мікроспоживання: інструкція, яка встановлює PCON.1=1 (PD), є останньою виконуваною командою перед переходом у режим мікроспоживання. У цьому режимі генератор, що задає, вимикається, припиняючи тим самим роботу всіх вузлів МК і зберігається тільки вміст ОЗП.

Єдиним виходом із цього стану є апаратне скидання RST. У цьому режимі роботи напруга U_{cc} може бути зменшена до 2 В і повинна бути відновлена до номінального перед виходом з режиму мікроспоживання.

Скидання слід утримувати в активному стані не менш 10 мс (час відновлення роботи задаючого генератора). При записі IDL=1 і PD=1 перевага має біт PD.

У табл. 1.17 представлені стани виводів МК у режимах холостого ходу і мікроспоживання.

Таблиця 1.17 - Стану виводів МК у режимах холостого ходу і мікроспоживання

Режим	Пам'ять програм	ALE	$\overline{PME}$	Порт P0	Порт P1	Порт P2	Порт P3
Холостого ходу	Внутрішня	1	1	Дані	Дані	Дані	Дані
Холостого ходу	Зовнішня	1	1	Z	Дані	Адреса	Дані
Мікроспоживання	Внутрішня	0	0	Дані	Дані	Дані	Дані
Мікроспоживання	Зовнішня	0	0	Z	Дані	Дані	Дані

Режим зниженого споживання для МК серії 1816 (n-МОН)

Під час нормальної роботи внутрішня ОЗП живиться від U_{cc} . Однак для МК серії 1816 сімейства МК51, якщо напруга на виводі RST перевищує U_{cc} , воно стає джерелом живлення для ОЗП. Це реалізовано за допомогою двох внутрішніх діодів, з катодів яких береться живлення ОЗП, а аноди підключені відповідно до входу RST

і до виводу живлення МК U_{cc} . Необхідно підкреслити, що для КМОП МК даний режим відсутній.

Логіка керування ЕОМ залежно від режиму роботи МК виробляє необхідний набір керуючих сигналів.

1.11 Система команд МК51

Мікроконтролер має 111 базових команд передачі даних, арифметичних і побітних операцій, передачі керування і операцій з бітами. Більшість команд виконуються за 1 або 2 машинних циклів при $f_{\text{так}}=12\text{МГц}$ і тривалості машинного циклу 1 мкс. Перший байт команди завжди містить код операції (КІП), другий і третій байти або адреси операндів, або безпосередні операнди. Перелік команд МК51 приводиться в таблиці 1.18.

Таблиця 1.18 - Система команд

Мнемокод	КОП	Мнемокод	КОП	Мнемокод	КОП
ACALL 0xxH	11	AJMP 5XXH	A1	DA A	D4
ACALL 1xxH	31	AJMP 6XXH	C1	DEC A	14
ACALL 2xxH	51	AJMP 7XXH	E1	DEC ad	15
ACALL 3xxH	71	ANL A, ad	55	DEC R0	18
ACALL 4xxH	91	ANL A, R0	58	DEC R1	19
ACALL 5xxH	B1	ANL A, R1	59	DEC R2	1A
ACALL 6xxH	D1	ANL A, R2	5A	DEC R3	1B
ACALL 7xxH	F1	ANL A, R3	5B	DEC R4	1C
ADD A, ad	25	ANL A, R4	5C	DEC R5	1D
ADD A, R0	28	ANL A, R5	5D	DEC R6	1E
ADD A, R1	29	ANL A, R6	5E	DEC R7	1F
ADD A, R2	2A	ANL A, R7	5F	DEC @R0	16
ADD A, R3	2B	ANL A, @R0	56	DEC @R1	17
ADD A, R4	2C	ANL A, @R1	57	DIV AB	84
ADD A, R5	2D	ANL A, #d	54	DJNZ ad, rel	D5
ADD A, R6	2E	ANL ad, A	52	DJNZ R0, rel	D8
ADD A, R7	2F	ANL ad, #d	S3	DJNZ R1, rel	D9
ADD A, @R0	26	ANL C, bit	82	DJNZ R2, rel	DA
ADD A, @R1	27	ANL C, /bit	BO	DJNZ R3, rel	DB

ADD A, #d	34	CJNE A, ad,	B5	DJNZ R4, rel	DC
ADDC A, ad	35	CJNE A, #d,	B4	DJNZ R5, rel	DD
ADDC A, R0	38	CJNE R0, #d,	B8	DJNZ R6, rel	DE
ADDC A, R0	39	CJNE R1, #d,	B9	DJNZ R7, rel	DF
ADDC A, R0	3A	CJNE R2, #d,	BA	INC a	04
ADDC A, R0	3B	CJNE R3, #d,	BB	INC ad	05
ADDC A, R0	3C	CJNE R4, #d,	BC	INC DPTR	A3
ADDC A, R0	3D	CJNE R5, #d,	BD	INC R0	08
ADDC A, R0	3E	CJNE R6, #d,	BE	INC R1	09
ADDC A, R0	3F	CJNE R7, #d,	BF	INC R2	0A
ADDC A, @R0	36	CJNE @R0,	B6	INC R3	0B
ADDC A, @R1	37	CJNE @R1,	B7	INC R4	0C
ADDC A, #d	24	CLR A	E4	INC R5	0D
AJMP 0XXH	01	CLR bit	C2	INC R6	0E
AJMP 1XXH	21	CLR C	C3	INC R7	0F
AJMP 2XXH	41	CPL A	F4	INC @R0	06
AJMP 3XXH	61	CPL bit	B2	INC @R1	
AJMP 4XXH	81	CPL C	B3	JB bit, rel	
				JBC bit, rel	
JC rel	40	MOV ad, @R0	86	MOV R7, ad	AF
JMP@A+DPTR	73	MOV ad, @R1	87	MOV R7, #d	7F
JNB bit, rel	30	MOV ad, #d	75	MOV @R0, A	F6
JNC rel	50	MOV ad, ads	85	MOV @R0, ad	A6
JNZrel	70	MOV bit, C	92	MOV @R0, #d	76
JZ rel	60	MOV C, bit	A2	MOV @R1, A	F7
LCALL ad16	12	MOV DPTR,	90	MOV @R1, ad	A7
LJMP ad 16	02	MOV R0, A	F8	MOV @R1, #d	77
MOV A, ad	E5	MOV R0, ad	A8	MOVC A, @+DPTR	93
MOV A, R0	E8	MOV R0, #d	78	MOVC A, @+PC	83
MOV A, R1	E9	MOV R1, A	F9	MOVX A, @DPTR	EO
MOV A, R2	EA	MOV R1, ad	A9	MOVX A, @R0	E2
MOV A, R3	EB	MOV R1, #d	79	MOVX A, @R1	E3
MOV A, R4	EC	MOV R2, A	FA	MOVX @DPTR, A	F0
MOV A, R5	ED	MOV R2, ad	AA	MOVX @R0, A	F2

MOV A , R6	EE	MOV R2, #d	7A	MOVX @R1, A	F3
MOV A , R7	EF	MOV R3 , A	FB	MUL AB	A4
MOV A , @R0	E6	MOV R3 , ad	AB	NOP	00
MOV A , @R1	E7	MOV R3 , #d	7B	ORL A , ad	45
MOV a , #d	74	MOV R4, A	FC	ORL A , R0	48
MOV ad , A	F5	MOV R4 , ad	AC	ORL A, R1	49
MOV ad , R0	88	MOV R4, #d	7C	ORL A, R2	4A
MOV ad , R1	89	MOV R5, A	FD	ORL A , R3	4B
MOV ad , R2	8A	MOV R5 , ad	AD	ORL A, R4	4C
MOV ad, R3	8B	MOV R5 , #d	7D	ORL A, R5	4D
MOV ad , R4	8C	MOV R6 , A	FE	ORL A, R6	4E
MOV ad , R5	8D	MOV R6, ad	AE	ORL A, R7	4F
MOV ad , R6	8E	MOV R6, #d	7E	ORL A , @R0	46
MOV ad , R7	8F	MOV R7 , A	FF	ORL A , @R0	47
ORL A , #d	44	RRC A	13	SUBB A , R7	9F
ORL ad , A	42	SETB bit	D2	SUBB A , @R0	
ORL ad , #d	43	setb c	D3	SUBB A , @R1	97
ORL C , bit	72	SJMP rel	80	SWAP A	C4
ORL C , /bit	AO	SUBB A, ad	95	XCH A , ad	C5
POP ad	DO	SUBB A, R0	98	XCH A, R0	C8
PUSH ad	CO	SUBB A, R1	99	XCH A, R1	C9
RET	22	SUBB A , R2	9A	XCH A , R2	CA
RETI	32	SUBB A , R3	9B	XCH A , R3	CB
RL A	23	SUBB A, R4	9C	XCH A , R4	CC
RLC A	33	SUBB A , R5	9D	XCH A , R5	CD
RR A	03	SUBB A, R6	9E	XCH A , R6	CE
XCH A, R7	CF	XRL A, R1	69	XRL A , R7	6F
XCH A , @R0	06	XRL A ,	6A	XRL A , @R0	66
XCH A , @R1	C7	XRL A, R3	6B	XRL A , @R1	67
XCHD A , @R0	D6	XRL A, R4	6C	XRL A , #d	64
XCHD A , @R1	D7	XRL A, R5	6D	XRL ad , A	62
XRL A, ad	65	XRL A , R6	6E	XRL ad , #d	63
XRL A, R0	68				

2 ЛАБОРАТОРНА РОБОТА №1

ЦИФРОАНАЛОГОВЕ ПЕРЕТВОРЕННЯ

Мета роботи: Вивчення методів цифроаналогового перетворення.

Учбове завдання: Розробка програм для формування різних аналогових сигналів.

2.1 Порядок виконання лабораторної роботи

2.1.1 Вивчити структурну схему модуля ЦАП на платі розширення.

2.1.2 Розробити алгоритм для виконання індивідуального завдання до початку лабораторного заняття.

2.1.3 Розробити програму для виконання індивідуального завдання до початку лабораторного заняття.

2.1.4 Ввести програму індивідуального завдання на персональному комп'ютері.

2.1.5 За допомогою ПОС проаналізувати виконання індивідуальної програми.

2.1.6 Завантажити програму в стенд ОЗВМ. Переконалися в правильному виконанні індивідуального завдання (формування заданої форми сигналу) використовуючи осцилограф, при негативному результаті здійснити зміну алгоритму або програми. Повторити завантаження програми в стенд ОЗВМ.

2.1.7 Роздрукувати лістинг правильно працюючої програми.

2.1.8 Відповісти на контрольні питання викладача.

2.2 Короткі теоретичні відомості

Цифро-аналогові перетворювачі (ЦАП) виробляють напругу або струм, функціонально пов'язані з керуючим кодом. Застосовуються ЦАП для формування вихідних аналогових сигналів цифрових вимірювальних і обчислювальних пристроїв. Для перетворення двійкового коду в аналоговий сигнал зазвичай формуються струми, пропорційні вагам розрядів коду, і потім підсумовуються ті із струмів, які відповідають ненульовим розрядам вхідного коду.

Застосовуються в основному два методи ЦАП: підсумовування одиничних еталонних величин і підсумовування еталонних величин, ваги яких розрізняються. У першому при формуванні вихідної аналогової величини використовується тільки одна еталонна величини вагою в один квант. У другому методі застосовуються еталонні величин з вагами, залежними від номера розряду, і в підсумовуванні беруть участь тільки ті еталонні величини, для яких у відповідному розряді вхідного коду встановлена одиниця.

У разі використання на вході двійкового позиційного коду значення усіх розрядів поступає одночасно, і робота таких ЦАП описується виразом:

$$X = P(a_1 \cdot 2^{E-1} + a_2 \cdot 2^{E-2} + a_3 \cdot 2^{E-3} + \dots + a_i \cdot 2^E - b) \quad (2.1)$$

де X - аналогова величина;

a_i - коефіцієнти відповідних двійкових розрядів, які набувають дискретних значень одиниця або нуль;

P - опорний сигнал;

b - число розрядів.

У перетворювачах з опорної напруги формуються еталонні величини, відповідні значенням розрядів вхідного коду, які підсумовуються і утворюють дискретні значення вихідної аналогової величини.

Класифікація ЦАП може бути проведена за ознаками:

- способом формування вихідної напруги (С підсумовування напруги, діленням напруги, підсумовуванням струмів);

- родом вихідного сигналу (із струмовим виходом, виходом по напрузі);

- полярністю вихідної напруги (постійному, змінному) і так далі.

Основні структури використовувані в ЦАП інтегрального виконання - це:

- структури з підсумовуванням струмів;

- ЦАП - із зваженими резисторами в ланцюгах емітерів;

- ЦАП - із зваженими резисторами в ланцюгах навантаження;

- ЦАП - зі сходовою матрицею $R=2R$ в ланцюгах емітерів транзисторів джерел струмів;

- ЦАП з вихідною сходовою матрицею $R=2R$.

2.3 Параметри ЦАП

Число розрядів n управляючого коду, номінальний вихідний струм, час встановлення вихідного сигналу після зміни вхідного управляючого коду, погрішність повної шкали, погрішність лінійності, диференціальна нелінійність.

Погрішності ЦАП можуть бути виражені у відсотках або інших відносних одиницях, а також в долях кванта.

Спрощена схема ЦАП для розуміння роботи перетворення вхідного коду в струм (напругу) певної величини наведена на рис. 2.1.

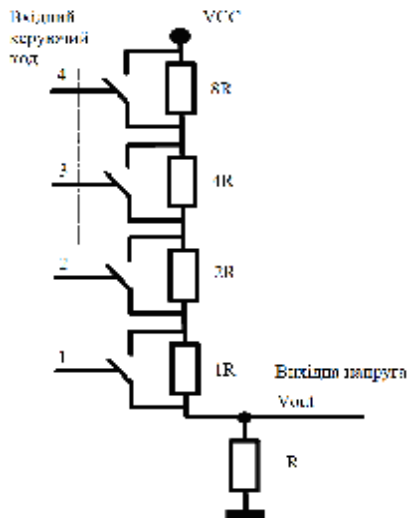


Рисунок 2.1 - Спрощена схема ЦАП

Якщо співвідношення резисторів ЦАП як 8R, 4R, 2R, 1R, то при включенні усіх комутаторів, відповідно до закону Ома напруга в точці OUT дорівнюватиме напрузі VCC. Якщо включити комутатор резистора 8R, напруга в точці OUT буде рівна. Аналогічно можна знати значення напруги, при різних комбінаціях вхідного коду.

У складі учбово-налагоджувального стенду є мікросхема ЦАП КР572ПА1. Вона включена як десяти розрядний ЦАП з виходом по напрузі (Напруга повної шкали вказується на платі розширення). Для спостереження вихідного сигналу з мікросхеми ЦАП необхідно підключити осцилограф до BNC рознімання і замкнути перемичку

J4. Цифровий код подається через лінії PA0-PA7 і PC0- PC1 мікросхеми паралельного приймача DD10.

2.4 Приклад виконання програми цифро-аналогового перетворення

Сформувати два прямокутні сигнали $\frac{1}{2}$ амплітуди і $\frac{1}{4}$ повної амплітуди виходу ЦАП.

A55 equ 8000h	; Адреса порту A
B55 equ 8001h	; Адреса порту B
C55 equ 8002h	; Адреса порту C
RUS55 equ 8003h	; Адреса порту керуючого слова
UW equ 80h	; Керуюче слово всі порти на вивід
ORG 0000	
MOV DPTR, #RUS55	
MOV A, #UW	
MOVX @DPTR, A	; ініціалізація BB55
M2: MOV R0, #00h	; в R0 зберігається старша частина
	; коду ЦАП
MOV DPTR, #C55	; скидання(обнулення) старших
MOV A, R0	; розрядів ЦАП
MOVX @DPTR, A	; D9 – D8
MOV DPTR, #A55	; скидання (обнулення)
MOV A, R0	; розрядів ЦАП
MOVX @DPTR, A	; D7 – D0
CALL ZAD	; підпрограма затримки на 0,1 сек
MOV A, #02h	; в A занести $\frac{1}{2}$ макс. значення коду
MOV DPTR, #C55	
MOVX @DPTR, A	; запис в порт A молодшої частини коду
CALL ZAD	; підпрограма затримки на 0,1 сек
MOV A, #01h	; в A занести $\frac{1}{2}$ макс. значення коду
MOV DPTR, #C55	
MOVX @DPTR, A	; запис в порт A молодшої частини коду
CALL ZAD	; підпрограма затримки на 0,1 сек
CALL ZAD	; підпрограма затримки на 0,1 сек
JMP M2	; перехід на новий цикл
END	

2.5 Завдання до лабораторної роботи

Таблиця 2.1 – Варіанти завдань до лабораторної роботи

№	Текст індивідуального завдання
1	Сформувати пілкоподібну напругу з частотою повторення 50 Гц. Відобразити на C_Інд число згенерованих імпульсів.
2	Після натиснення S1 сформувати трикутні імпульси, передній фронт 20 мсек, задній 10 мсек.
3	Після натиснення S2 сформувати трапецевидні імпульси, передній фронт 13 мсек, задній 15 мсек.
4	Сформувати синусоїду з частотою повторення 120 Гц.
5	Сформувати пілкоподібну напругу з частотою повторення 200 Гц і тривалістю переднього фронту 2 мсек.
6	Після натиснення S3 сформувати синусоїду з частотою повторення 100 Гц.
7	Сформувати прямокутні імпульси, з тривалістю 25мсек і шпаруватістю 4.
8	Після натиснення S4 сформувати трикутні імпульси, передній фронт 25 мсек, задній 5 мсек.
9	Сформувати синусоїду з частотою повторення 300 Гц. Після натиснення S5 змінити частоту на 100 Гц.
10	Сформувати два прямокутні імпульси, один максимальною амплітудною тривалістю і другий 2/3 максимальної амплітуди з періодом повторення 40 Гц.
11	Сформувати прямокутні імпульси, з тривалістю 25 мсек після натиснення S6 сформувати трикутні імпульси.
12	Сформувати синусоїду з частотою повторення 70 Гц, по натисненню S7 прямокутні імпульси, з тривалістю 25 мсек і шпаруватістю 2.
13	Після натиснення S8 сформувати трикутні імпульси, передній фронт 15 мсек, задній 40 мсек.
14	Після натиснення S9 сформувати трапецевидні імпульси, передній фронт 20 сек, задній 20 мсек.
15	Сформувати три прямокутні імпульси, один 1/3 максимальних амплітуди, другий 2/3 і третій максимальною амплітудою з періодом повторень 100 Гц.

2.6 Контрольні питання

- 1 Методи і типи ЦАП.
- 2 Статичні параметри ЦАП.
- 3 Поняття дискретності, квантування, роздільна здатність ЦАП.
- 4 Характеристика перетворення, нелінійність ЦАП.
- 5 Напруга зміщення нуля, допустима напруга на виході ЦАП.
- 6 Динамічні параметри ЦАП.
- 7 Чинники, що впливають на погрішність ЦАП.
- 8 Апаратні реалізації ЦАП.
- 9 Приклади практичного застосування ЦАП.

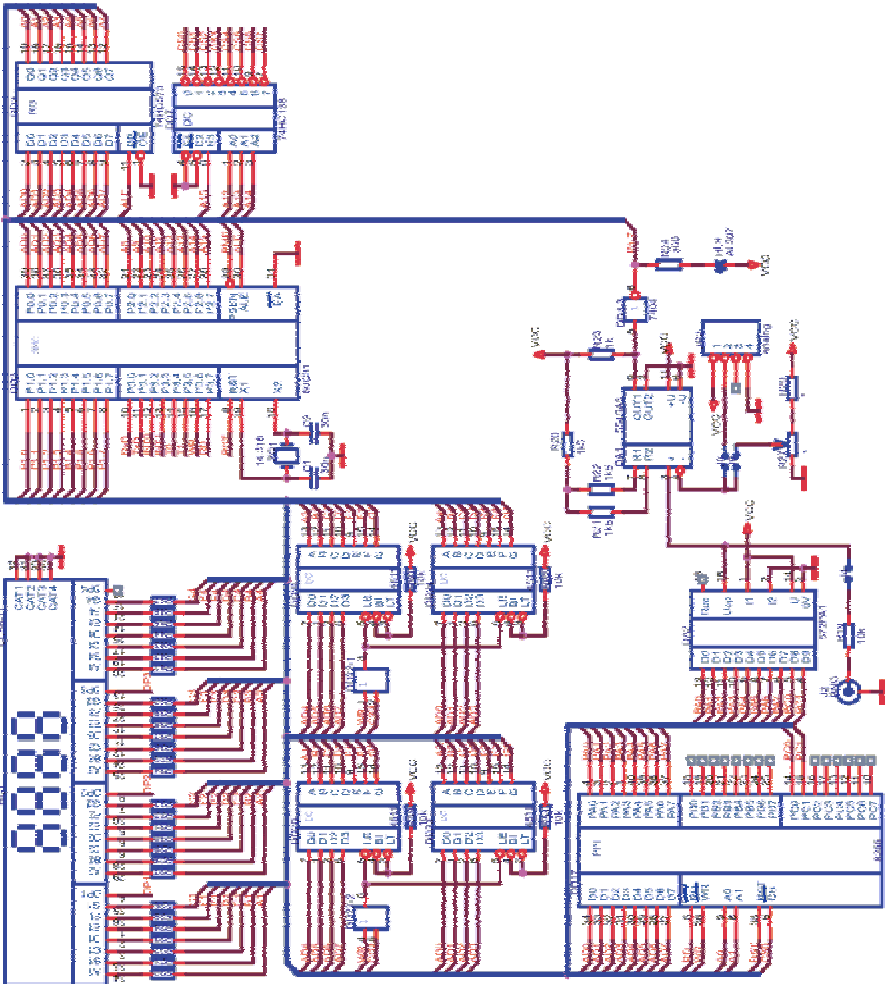


Рисунок 2.2 - EV8031

3 ЛАБОРАТОРНА РОБОТА №2

АНАЛОГО-ЦИФРОВЕ ПЕРЕТВОРЕННЯ

Мета роботи: Навчитися вимірювати аналогову величину.

Учбове завдання: Розробка програм вимірювання аналогових величин для різних методів вимірювання і типів АЦП.

3.1 Порядок виконання лабораторної роботи:

3.1.1 Вивчити структурну схему модуля АЦП на платі розширення.

3.1.2 Розробити алгоритм та програму для виконання індивідуального завдання до початку лабораторного заняття.

3.1.3 Ввести програму індивідуального завдання на персональному комп'ютері.

3.1.4 За допомогою ПОС проаналізувати виконання індивідуальної програми.

3.1.5 Завантажити програму в стенд ОЕВМ. Переконалися в правильному виконанні індивідуального завдання, змінити значення напруги, які подаються на вхід АЦП, повторити перетворення, при негативному результаті здійснити зміну алгоритму або програми.

3.1.6 Повторити завантаження програми в стенд ОЕВМ.

3.1.7 Роздрукувати лістинг правильно працюючої програми.

3.1.8 Відповісти на контрольні питання викладача.

3.2 Короткі теоретичні відомості

Аналогово-цифрові перетворювачі (АЦП) застосовуються у вимірювальних системах і вимірювально-обчислювальних комплексах для узгодження аналогових джерел вимірюваних сигналів з цифровими пристроями обробки і представлення результатів вимірювання.

Існують різні методи побудови АЦП. Вони відрізняються за складністю реалізації, завадостійкістю, швидкодією.

У системах де основним критерієм є швидкодія, застосовують АЦП паралельного перетворення. Але АЦП цього типу досить складні

в реалізації. Для n -розрядного АЦП необхідно $2^n - 1$ компараторів і паралельний дільник напруги, який виробляє $2^n - 1$ рівнів квантування.

Для реалізації систем з високою перешкодостійкістю застосовують інтегруючі АЦП. Такий АЦП складається з двох перетворювачів. Вимірювана напруга перетворюється на тривалість імпульсу, а потім тривалість імпульсу перетворюється в цифровий код.

Одним з найпоширеніших є АЦП, побудований на цифро-аналоговому (ЦАП) перетворювачі. Схема цього АЦП наведена на рисунку 3.1.

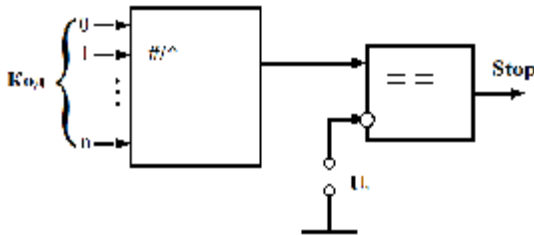


Рисунок 3.1 - Схема АЦП побудованого на ЦАП.

Код формується лічильником, при організації жорсткої логіки, або програмно, якщо АЦП працює у складі обчислювального комплексу. Вхідний код перетворюється в аналоговий сигнал за допомогою ЦАП. Напруга з виходу ЦАП надходить на один з входів компаратора. На другий вхід подається вимірюєма напруга U_x . У момент, коли напруга ЦАП буде рівна вимірюваній, компаратор формує сигнал “Stop” який свідчить про закінчення циклу вимірювання.

При формуванні коду використовуються різні алгоритми. Найпростішим алгоритмом є порозрядне врівноваження. При такому підході код змінюється від мінімального, шляхом збільшення одиниці молодшого розряду до тих пір, поки напруга ЦАП не зрівняється з вимірюваною напругою. Недоліком порозрядного врівноваження є маленьке швидкодію.

Для скорочення часу перетворення застосовується метод половинних наближень. Урівноваження починається зі

старшого розряду. У цьому розряді встановлюється одиниця і читається стан компаратора. Якщо напруга ЦАП більше вимірюваної то розряд скидається, а якщо менше, то розряд зберігає свій стан.

Далі таким же чином обробляється наступний розряд. Перетворення закінчується тоді, коли буде опрацьовано всі розряди.

У системах стеження, за будь-якими параметрами часто необхідно безперервно зчитувати стан датчика. Це забезпечується малим часом перетворення за рахунок застосування слідкуючого АЦП. Суть даного алгоритму полягає в тому, що спочатку код формується методом половинних наближень. А після порівняння з вимірюваною напругою АЦП відстежує зміну напруги. Якщо напруга зростає то код збільшується за розрядами до тих пір, поки напруга ЦАП не зрівняється з вимірюваною, і навпаки.

АЦП побудований на мікросхемах DA2 (десяти розрядний ЦАП) і DA1 компаратор з ТТЛ виходом. Молодша частина коду зберігається у порту А мікросхеми паралельного інтерфейсу 580BB55 (див. схему стенда), старша частина коду зберігається в розрядах 0-1 порту С. Стан компаратора можна прочитати на ввіді порту P1.7 однокристалної ЕОМ, про закінчення циклу перетворення також свідчить світіння світлодіода HL9. Вимірювана напруга знімається з дільника побудованого на резисторах P.27, R28. Обертаючи ручку потенціометра P.27 можна міняти вимірювану напругу.

3.3 Приклад реалізації алгоритму порозрядного врівноваження

```
$nolist
#include (c:\asm51\compile\mod51)
$list
; програма аналого-цифрового перетворення
A55 equ 8000h          ; Адреса порту А
B55 equ 8001h          ; Адреса порту В
C55 equ 8002h          ; Адреса порту С
RUS55 equ 8003h        ; Адреса порту керуючого слова
UW equ 80h             ; Керуюче слово всі порти на вивід
ORG 0000
MOV DPTR, #RUS55
MOV A, #UW
MOVX @DPTR, A          ; ініціалізація BB55
```

```

M2: MOV R0, #00b      ; в R0 зберігається старша частина
                       ; коду ЦАП
MOV DPTR, #C55        ; скидання(обнулення) старших
MOV A, R0              ; розрядів ЦАП
MOVX @DPTR, A         ; D9 – D8
M4: MOV A, #00h        ; в A зберігається молодша частина
                       ; коду ЦАП
MOV DPTR, #A55        ; скидання(обнулення) молодших
                       ; розрядів ЦАП
MOVX @DPTR, A         ; D7 – D0
M1: INC A              ; збільшення на 1 молодшої частини коду
MOV DPTR, #A55        ; Запис в порт A молодшу частину коду
MOVX @DPTR, A         ; Аналіз стану компаратору, якщо 0
JNB P1.7, M3           ; то перехід до відображення коду мітки M3
CJNE A, #0FFh, M1      ; якщо молодша частина коду FFh то
                       ; перехід до збільшення старшої частини коду
INC R0                ; інакше перехід на мітку M1
MOV DPTR, #C55
MOV A, R0
MOVX @DPTR, A         ; Запис в порт C старшу частину коду
CNJE R0, #03h, M4      ; якщо старша частина коду 03h то
JMP M2                ; перехід до мітки M4 інакше до мітки M2
M3: MOV DPTR, #0B000h  ; Вивід на індикацію
MOVX @DPTR, A         ; молодшої частини коду
MOV A, R0
MOV DPTR, #0A000h     ; Вивід на індикацію
MOVX @DPTR, A         ; старшої частини коду
MOV R2, #0FFh         ; Часова затримка реалізована
M5: MOV R2, #0FFh     ; на двох регістрах з декрементуванням
DJNZ R3,$              ; у вкладених циклах
DJNZ R2, M5
JMP M2                ; перехід на новий цикл вимірювання
END

```

3.4 Завдання до лабораторної роботи

Таблиця 3.1 - Завдання до лабораторної роботи

№	Алгоритм для реалізації
1	Порозрядне урівноваження
2	Метод половинних наближень
3	Що стежить з первинним порозрядним урівноваженням
4	Що стежить з первинним половинним наближенням
5	Після натиснення кнопки S1 запустити АЦП половинних наближень
6	Після натиснення кнопки S2 запустити АЦП порозрядного урівноваження
7	Після натиснення кнопки S3 запустити АЦП типу, що стежить з первинним порозрядним урівноваженням
8	Після натиснення кнопки S4 запустити АЦП типу, що стежить з первинним половинним наближенням
9	Після натиснення кнопки S5 запустити АЦП половинних наближень, результат відобразити по натисненню кнопки S2
10	Після натиснення кнопки S6 запустити АЦП порозрядного урівноваження, результат відобразити по натисненню кнопки S2
11	Після натиснення кнопки S7 запустити АЦП типу, що стежить з первинним порозрядним урівноваженням, результат відобразити по натисненню кнопки S8
12	Після натиснення кнопки S8 запустити АЦП типу, що стежить з первинним половинним наближенням результату, після натиснення кнопки S9

3.5 Контрольні питання

1. Методи і типи АЦП.
2. Статичні параметри АЦП.
3. Поняття дискретності, квантування, роздільна здатність.
4. Характеристика перетворення, диференційна нелінійність АЦП, відхилення коефіцієнта перетворення.
5. Напруга зсуву нуля.
6. Динамічні параметри АЦП.
7. Час перетворення, час затримки запуску, час циклу перетворення, максимальна частота перетворення.
8. Поняття апертурний час.
9. Фактори, що впливають на похибку АЦП.
10. Апаратні реалізації АЦП.

11 Приклади практичного застосування АЦП.

12 Побудова схем АЦП за допомогою мікросхем ЦАП.

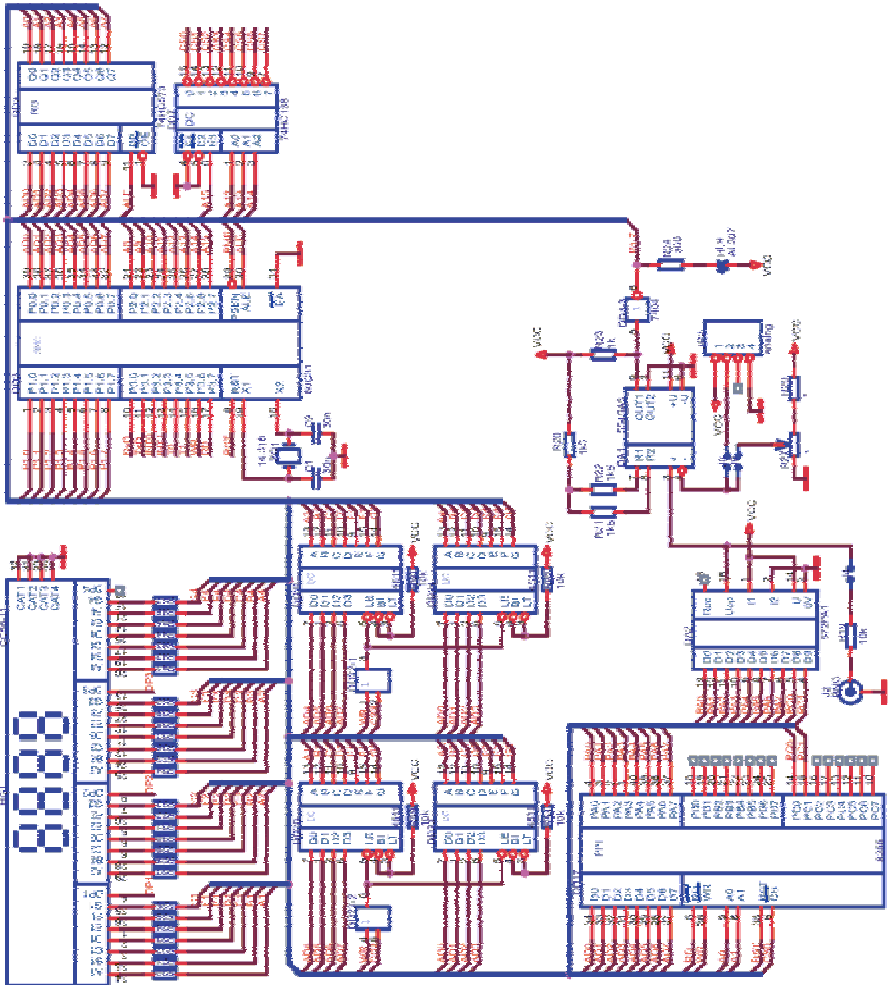


Рисунок 3.2 - EV8031

4 ЛАБОРАТОРНА РОБОТА № 3

ОБРОБКА Й ФОРМУВАННЯ СИГНАЛІВ КЕРУВАННЯ ВИКОНАВЧИМИ МЕХАНІЗМАМИ. РОЗРОБКА ПРОГРАМИ ПІДРАХУНКУ ЧИСЛА СПРАЦЬОВУВАНЬ ДАТЧИКА, ФОРМУВАННЯ ІМПУЛЬСНИХ І СТАТИЧНИХ СИГНАЛІВ КЕРУВАННЯ

Мета роботи: освоїти основні методи аналізу вхідних сигналів від імпульсних джерел у мікроконтролер КР1816ВЕ51 та програмної генерації їм динамічних, статичних керуючих вихідних сигналів для виконавчих механізмів у середовищі «Proteus» (demo-версія).

Перед початком роботи слід ознайомитися із середовищем «Proteus» [1-3] і її можливостями.

Завдання має 2 розділи (непарний і парний варіант номера підгрупи):

- підрахунок числа вхідних імпульсів між двома подіями (непарний-1.1,2.1) або за заданий проміжок часу (парний-1.2,2.2);
- формування вихідних динамічних (непарний-1.1,2.1) або статичних (парний-1.2,2.2) сигналів керування.

4.1 Короткі теоретичні відомості

Зазвичай в керуючих програмах виникає необхідність очікування ланцюжка подій, що представляється послідовністю імпульсних сигналів від датчиків і формування відповіді на них. Розглянемо дві типові процедури: підрахунок числа імпульсів між двома подіями, підрахунок числа імпульсів за заданий інтервал часу.

4.1.1 Підрахунок числа імпульсів між двома подіями

Цю типову процедуру зручно проілюструвати на конкретному прикладі. Припустимо, що необхідно підраховувати число деталей, які зійшли з конвеєра від моменту його включення до моменту вимикання. Факт того, що деталь зійшла з конвеєра фіксується фотоелементом, на виході якого формується імпульсний сигнал, який фіксується КР1816ВЕ51. Особливість процедури очікування імпульсного сигналу полягає в тому, що повинен виявити не тільки факт появи, але й факт закінчення сигналу (відсутність цієї події може вказувати на те, що деталь «застрягла» на конвеєрі, відповідно

видати сигнал «зупинки конвеєра» з фіксацією часу події в регістрі). По закінченню виконання процедури в акумуляторі фіксується число деталей, представлене у двійково-десятковому (або ін. коді).

Для простоти реалізації програми вважаємо, що загальна кількість деталей не перевищує 250, в акумуляторі фіксується число деталей, представлене у двійковому коді (максимальна кількість деталей 255). На початку роботи конвеєра слід подати сигнал «включити конвеєр», а перед опитуванням лічильника «виключити конвеєр».

	; версія для МК51
mov tmod,#01000000b	; настроювання лічильника 1
mov th1,#0	; скидання лічильника деталей
wait0: jb p3.4,wait0	; очікування включення конвеєра
setb tcon.6	; пуск лічильника 1
wait1: jnb p3.4,wait1	; очікування вимикання конвеєра
clr tcon.6	; останов лічильника 1
mov a,th1	; (аккумулятор) <-- число
	;деталей
exit: ...	; вихід із процедури

4.1.2 Підрахунок числа імпульсів за заданий проміжок часу

При виконанні завдання, перетворення число-імпульсного коду у двійковий код, а також у ряді інших завдань може виникнути необхідність підрахунку числа імпульсів за заданий інтервал часу. Ця процедура може бути реалізовано чотирма різними способами:

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

Для МК51, що має у своєму складі два таймери/лічильники, можливий апаратний спосіб реалізації обох процесів: формування тимчасового інтервалу на Т/Л0 і підрахунок числа імпульсів на Т/Л1. Датчик імпульсів повинен бути підключений до входу Т1:

; версія для МК51

time equ not(10000)+1	; визначення константи для
	; відліку інтервалу 10 мс
mov tmod,#01010001b	; настроювання т/л, 16 біт
	; 1 - лічильник
	; 0 - таймер
clr a	; скидання акумулятора
mov th1,a	; скидання т/л1
mov tl1,a	
mov th0,#high(time)	; завантаження в т/л0
mov tl0,#low(time)	; константи time
orl tcon,#50h	; пуск т/л1 і т/л0
wait:jbc tcon.5,exit	; перевірка переповнення т/л0
sjmp wait	; цикл, якщо tf=0
exit:mov b,th1	; (b) (a)<--число імпульсів за 10 мс
mov a,tl1	
...	; вихід із процедури

4.1.3 Формування вихідних динамічних керуючих впливів

Сформувати на одному з виводів порту P2 мікроконтролера КР1816ВЕ51 періодичний імпульсний сигнал, період якого й шпаруватість задаються вмістом і-го регістру (див. варіант завдання).

Принцип роботи програми: зовнішня програма у випадкові моменти часу встановлює в регістри Ri мікроконтролера значення, що визначають тривалість імпульсу, період проходження сигналу і його шпаруватість. Підпрограма, що розроблюється повинна використовувати ці значення для формування сигналів на виходах портів мікроконтролера.

Генерація періодичного керуючого впливу (меандру). Для генерації меандру зручно скористатися процедурою видачі періодичного імпульсного керуючого впливу.

; версія для мк 51

```
meandr:
xcor: cpl p1.3
acall dlyx
sjmp xcor
```

Нескінченний (або кінцевий за умовою припинення події MEANDR) періодичний сигнал формується в лінії 3 порту 1. На інших лініях порту 1 сигнали залишаються незмінними.

4.1.4 Формування вихідних статичних сигналів керування

Формування статичних сигналів: для керування виконавчим механізмом, що працюють за принципом «ввімкнути» і «вимкнути», на відповідній вхідній лінії порту МК необхідно сформувати статичний сигнал 0 («вимкнути») або 1 («ввімкнути»), що реалізується командами виводу безпосереднього операнду, що містить у необхідному біті значення 0 або 1.

У випадку паралельного керування групою автономних виконавчих механізмів, підключених до вихідного порту, формується не двійковий керуючий вплив, а керуюче слово (ВУС), що має формат байта, кожному розряду якого ставиться у відповідність 1 або 0 залежно від того, які виконавчі механізми повинні бути включені, а які виключені. Керуючі слова зручно формувати командами логічних операцій над умістом порту. Команда ANL використовується для скидання тих біт ВУС, які в операнді (масці) задані нулем. Команда ORL використовується для установки біт ВУС. Командою XRL здійснюється інверсія біт відповідно до вираження $x + 1 = \sim x$ (тобто x інвертоване).

Приклад: до виводів портів P1 і P2 підключені виконавчі механізми. Поточний стан виконавчих механізмів зафіксовано в поточному слові стану системи. Зовнішня програма може змінити в довільні моменти часу еталонне слово стану системи.

Принцип роботи програми наступний:

- періодично, із заданим періодом, програма виконує запит еталонного слова стану (те, яке потрібно встановити) системи й порівнює його з поточним словом стану системи;
- при виявленні розбіжності змісту еталонного й поточного слів стану, програма змінює стан відповідних ліній порту, починаючи з молодших;
- кожна зміна стану виконавчих механізмів фіксується в поточнім слові стану системи;
- період опитування - 100 ms.

Програма повинна бути оформлена у вигляді підпрограми, запит до якої здійснюється командою CALL.

Для формування складних послідовностей ВУС зручно користуватися табличним методом, при якому усі можливі ВУС упаковані в таблицю, а прикладна програма МК обчислює адресу необхідного еталонного ВУС, вибирає його з таблиці й передає в порт виводу.

Формування аперіодичного керуючого сигналу:
 послідовність імпульсних сигналів з довільною тривалістю й шпаруватістю може бути отримана аналогічним чином, тобто шляхом чергування процедур видачі змінюваного значення сигналу (0 або 1) і виклику підпрограм тимчасових затримок заданої тривалості.

У МК 51 на вхід таймера/лічильника (Т/Л) можуть надходити сигнали синхронізації із частотою 1 МГц (Т/Л у режимі таймера) або сигнали від зовнішнього джерела (Т/Л у режимі лічильника). Обидва ці режими можуть бути використані для формування затримок. Якщо використовувати Т/Л у режимі таймера повного формату (16 біт), то можна одержати затримки в діапазоні 1 - 65 536 мкс.

Як приклад розглянемо організацію тимчасової затримки тривалістю 50 мс у МК 51. Передбачається, що біт ІЕ.7 встановлений.

; версія для мк 51

;організація переходу до мітки next при переповненні т/л0

```
org 0bh                ; адреса вектора переривання від т/л0
clr tcon.4             ; зупинка т/л0
reti                   ; вихід з підпрограми
                        ; обробки переривання
org 100h               ; початкова адреса програми
mov tmod,#01h          ; настроювання т/л0
mov tl0,#low(not(50000-1)) ; завантаження таймера
mov th0,#high(not(50000-1))
setb tcon.4            ; старт т/л0
setb ie.1              ; дозвіл переривання від т/л0
setb pcon.0            ; перехід мк 51 в режим холостого ходу
next: . . . . .
```

Керуючий вплив типу "одиначний імпульс" можна одержати послідовною видачею сигналів «ввімкнути» і «вимкнути» із проміжним викликом підпрограми тимчасової затримки. Тривалість імпульсу визначається тимчасовою затримкою, реалізованою підпрограмою DELAY.

```
puls:                  ;видача імпульсу в лінію 3 порту 1
on: anl p1,#11110111b ;включення виконавчого механізму
call delay             ;тимчасова затримка
off: orl p1,#00001000b ;відключення виконавчого механізму
```

4.2 Завдання до лабораторної роботи

Варіанти завдань по групах розділів 1.1,1.2:

- тривалість імпульсу 100 мкс, період проходження сигналу- 500мкс, шпаруватість 1:5.
- константа часу 50 мс, тривалість імпульсу 100 мкс, період проходження 200 мкс, шпаруватість 1:2
- тривалість імпульсу 250 мкс, період проходження сигналу - 1000мкс, шпаруватість 1:4.
- константа часу 100 мс, тривалість імпульсу 250 мкс, період проходження 500 мкс, шпаруватість 1:2

4.3 Порядок виконання лабораторних робіт із застосуванням середовища «Proteus» (demo- версія)

На першому занятті студент повинен ознайомитися з можливостями й особливостями роботи в середовищі «Proteus», завданням ЛР і побудувати на «робочому столі» електричну схему пристрою (по варіантах) на основі МП КР1816ВЕ51 (і80С51), випробувати і перевірити її функціонування й тестові програми (наведені в ЛР).

При самостійній підготовці до роботи студент повинен написати й відкоригувати власні програми відповідно до завдання (використовувати засоби «Proteus»), виконати необхідні контрольні розрахунки й провести моделювання в середовищі «Proteus».

На занятті студент повинен пред'явити їхньому викладачеві й одержати допуск до виконання роботи у ПК лабораторії, остаточно налагодити програму на ПК і, при необхідності, пояснити викладачеві роботу програми, її частин і обґрунтувати правильність її функціонування відповідно до варіанту завдання. Після цього робота вважається виконаною, викладач підписує чернетку виконаної ЛР.

До наступної ЛР студент повинен оформити звіт по виконаній роботі (на основі чорнового варіанта) і захистити його у викладача. Позитивна оцінка й підпис звіту викладачем означає, що ЛР здана.

Рекомендується підписані звіти зберігати до кінця навчання за курсом ВР на іспиті дозволяється користуватися студентам своїми звітами по ЛР при відповіді на запитання й розв'язку завдань.

5 ЛАБОРАТОРНА РОБОТА № 4

АРИФМЕТИЧНІ ОПЕРАЦІЇ І РОЗРОБКА ПРОГРАМИ ЇХ ВИКОНАННЯ

Мета роботи: освоїти основні методи аналізу вхідних даних і виконання типових алгоритмів арифметичних операцій з ними у форматі з фіксованої комою на основі мікроконтролера KP1816BE51 (80C51) у середовищі «Proteus» (demo- версія).

Перед початком роботи слід ознайомитися із середовищем «Proteus» [1-3] і її можливостями.

5.1 Короткі теоретичні відомості

Усі порти (0,1,2,3) МК-51 служать для побайтного введення - виводу даних. Крім того, порти 0 і 2 служать для виводу адреси. Виводи порту 3 можуть бути використані для реалізації альтернативних функцій. Порт 0 є двонаправленим, порти 1, 2, 3 - квазідвонаправленими: кожна лінія порту може бути налаштована для введення або виводу даних. Навантажувальна здатність портів 1, 2, 3 - один ТТЛ - вхід, а порту 0 - два ТТЛ входу. РС - 16-розрядний програмний лічильник.

Акумулятор (А) служить для зберігання як операнду, так і результату операції. Але МК-51 може виконувати ряд команд і без участі акумулятора. Виконання багатьох команд проводиться в арифметико-логічному пристрої, а ряд ознак операцій фіксується в регістрі слова стану програми (PSW).

Арифметико-логічний пристрій (8-бітний) може виконувати арифметичні операції додавання, віднімання, множення й ділення; логічні операції І, АБО, що виключає АБО, а також операції циклічного зрушення, скидання, інвертування і т.д. В АЛУ є програмно недоступні регістри Т1 і Т2, призначені для тимчасового зберігання операндів, схема десяткової корекції й схема формування ознак.

Восьмирозрядний показчик стека (SP) може адресувати будь-яку область пам'яті даних. Вміст його зменшується на одиницю в ході виконання команд PUSH і CALL і збільшується на одиницю при виконанні команд POP і RET. При скиданні в SP автоматично завантажуються початковий код 07 Н. 16-розрядний регістр-показчик даних (DPTR) служить для фіксації адреси при звертанні до зовнішньої пам'яті. Він може працювати як два незалежні 8-розрядних регістри DPH і DPL.

Команди пересилання даних. Як було розглянуто раніше, арифметичні й логічні команди можуть бути виконані тільки над

вмістом регістру акумулятора, тому винятково важливе значення в системі команд здобувають команди пересилання даних. За допомогою цих команд можна скопіювати вміст будь-якої комірки пам'яті в регістр-акумулятор або навпаки скопіювати вміст акумулятора в будь-яку комірку пам'яті. Тому що в мікроконтролері присутні три незалежні області пам'яті, то для звертання до них уведено різні команди:

- копіювання даних у внутрішньому ОЗП: MOV;
- обмін даними акумулятора із внутрішнім ОЗП: XCH, XCHD;
- копіювання із зовнішньої пам'яті даних: MOVX;
- копіювання даних з пам'яті програм: MOVC.

Будь-яке гніздо 256-байтового блоку внутрішнього ОЗП даних може бути обрана з використанням непрямо-регістрової адресації через регістри покажчики R0 і R1 (обраного банку робочих регістрів). Команди пересилання із прямою адресацією між комірками пам'яті дозволяють зтягати вміст порту в гніздо внутрішнього ОЗП або обмінюватися вмістом гнізд внутрішнього ОЗП між собою без використання акумулятора. Таблиці символів (кодів), записані в ПЗП програми можуть бути скопійовані в акумулятор за допомогою команд передачі даних з непрямою адресацією. Гніздо адресного простору 64 Кбайт зовнішнього ОЗП також може бути обрана з використанням непрямо-регістрової адресації через регістр покажчик даних DPTR. Вміст акумулятора може бути виміняне із вмістом робочих регістрів обраного банку.

Команди розгалуження й передачі керування. Команди розгалуження дозволяють реалізовувати умовні оператори й оператори циклів. У мікроконтролерах сімейства MCS-51 доступні наступні команди:

- безумовний перехід: LJMP, AJMP, SJMP;
- виклик і повернення з підпрограми: LCALL, ACALL, RET, RETI;
- перевірка вмісту акумулятора: JZ, JNZ, CJNE, JMP;
- перевірка прапора переносу 3: JC, JNC;
- перевірка вмісту будь-якого біта в бітовому просторі: JB, JNB, JBC.

Команди 16-розрядних безумовних переходів і викликів підпрограм дозволяють здійснити перехід у будь-яку точку адресного простору пам'яті програм обсягом до 64 Кбайт. Приклади команд:

Команди 11-розрядних переходів і викликів підпрограм дозволяють скоротити обсяг програми, але при цьому забезпечують переходи тільки усередині програмного модуля.

В системі команд є команди умовних і безумовних переходів щодо початкової адреси наступної команди в межах від (PC)-127 до (PC)+127.

Команди перевірки вмісту акумулятора й прапора переносу С можуть бути використані для реалізації перевірки різних умов. При цьому вміст не змінюється, тобто якщо потрібно зробити кілька перевірок однієї й тієї ж змінної, то повторно переносити значення цієї змінної в акумулятор не потрібно.

Як видно з наведених прикладів, команди переходів цього мікроконтролера дозволяють реалізувати набагато більш ефективні по кількості команд програми в порівнянні з іншими процесорами, такими як, наприклад MCS-48.

Непрямий перехід JMP @A+DPTR у системі команд мікроконтролерів сімейства MCS-51 забезпечує розгалуження програми по вмісту акумулятора А. Це дозволяє реалізовувати операцію переходу за заданим кодом, еквівалентну операторові case у мові програмування pascal, але набагато швидше (за два машинні цикли). Використання в цій команді покажчика даних DPTR дозволяє розміщати таблицю переходів у будь-якому місці пам'яті програм.

Арифметичні команди. У наборі команд мікроконтролера є наступні арифметичні операції:

- додавання ADD;
- додавання з урахуванням прапора переносу ADDC;
- віднімання з запозиченням SUBB;
- інкрементування (збільшення на 1) INC;
- декрементування (зменшення на 1) DEC;
- десяткова корекція DA;
- множення MUL;
- ділення DIV.

Дії проводяться над цілими числами без знака. При операції множення вміст акумулятора А множиться на вміст регістру В, і результат розташовується таким способом: молодший байт у регістрі В, старший - у регістрі А. При виконанні операції ділення, ціле від ділення поміщується в акумулятор А, залишок у регістр В.

Приклади схем алгоритмів множення на двійковому суматорі прямого коду (ДСПК): множене й множник - цілі числа - а; множене - ціле число, множник - дробове число - б. множене й множник - цілі числа - а; множене - ціле число, множник - дробове число - б.

Знак одержуваного добутку визначається до його проведення по формулі:

Далі наведені приклади схем алгоритмів на рис.5.1-5.3: додавання цілих чисел зі знаком на двійковому суматорі доповняльного коду (ДСДК); переводу чисел: із прямого коду в доповняльний код ; з доповняльного коду в прямий код.

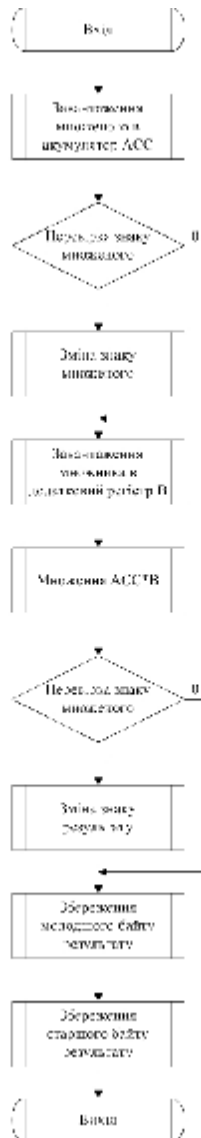


Рисунок 5.1 - Додавання цілих чисел зі знаком на двійковому суматорі доповняльного коду (ДСДК)

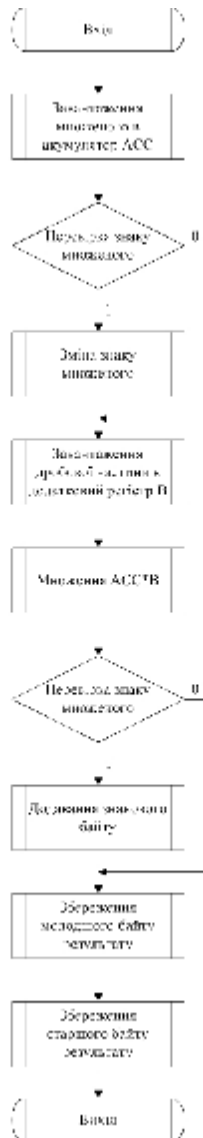


Рисунок 5.2 - Перевід чисел: із прямого коду в доповняльний код

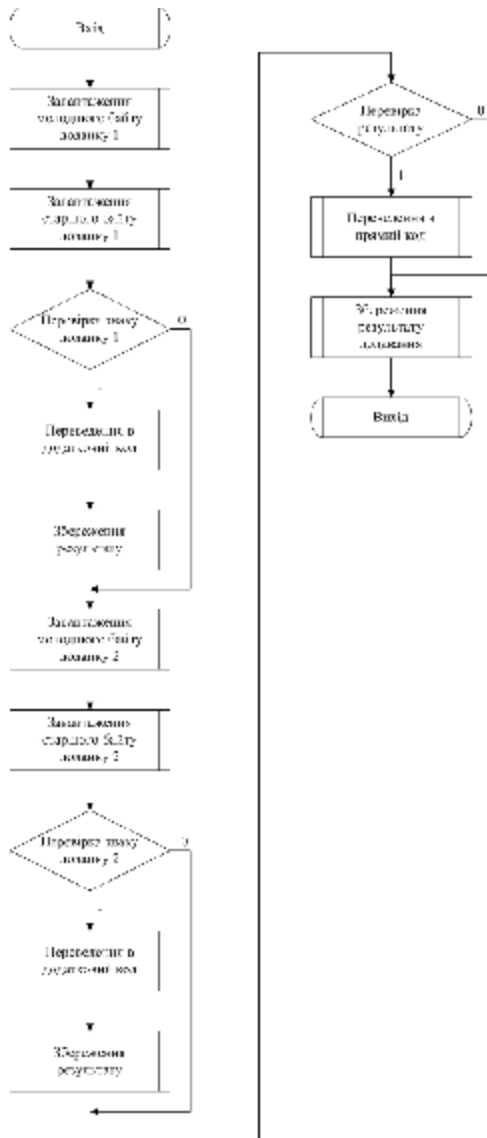
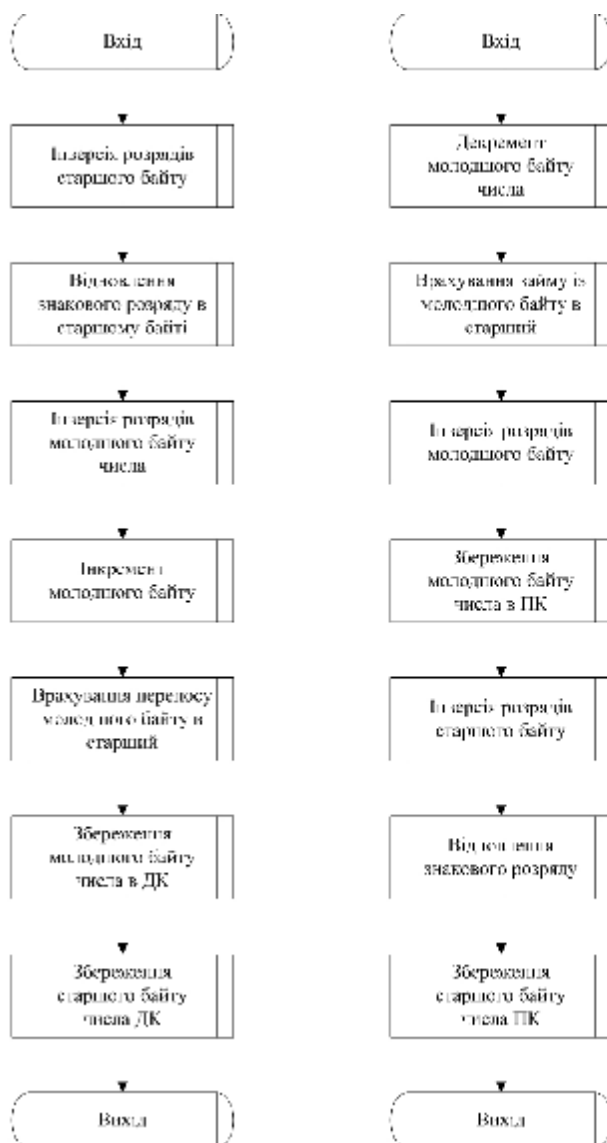


Рисунок 5.3 - Перевід чисел з доповняльного коду в прямий код.



5.2 Завдання до лабораторної роботи

5.2.1 Перше завдання це скласти два 8-розрядних числа та розмістити результат згідно з варіантом табл..5.1-5.2.

Таблиця 5.1 - Скласти два 8-розрядних числа X и Y (додавання на ДСДК)

Варіант	X	Y
I	1EH	24H
II	1FH	35H
III	5DH	20H
IV	25H	7BH
V	8AH	CBH

Таблиця 5.2 - Результат помістити

Варіант	Внутрішнє ОЗП	Внутрішнє ОЗП	Порт
I	06H	20H	P2
II	4BH	68H	P0
III	72H	2AH	P3
IV	3EH	B5H	P1
V	54H	D0H	P2

5.2.2 Помножити два 8-розрядних числа та розмістити результат згідно з варіантом табл..5.3-5.4.

Таблиця 5.3 - Помножити два 8-розрядних числа X и Y

Варіант	X	Y
I	A0H	09H
II	2BH	7AH
III	1FH	6DH
IV	83H	E4H
V	37H	FAH

Таблиця 5.4 - Результат помістити

Варіант	Внутрішнє ОЗП	Порти
I	18H	P2,P3
II	0FH	P0, P1
III	3AH	P1, P3
IV	6DH	P2, P3
V	7AH	P0, P1

5.2.3 Зробити вирахування (через додавання на ДСДК) 16-розрядних чисел A и B та розмістити результат згідно з варіантом табл..5.5-5.6.

Таблиця 5.5 - Зробити вирахування (через додавання на ДСДК) 16-розрядних чисел А и В

Варіант	А	В
I	201FH	0145H
II	7A04H	150AH
III	9F1DH	70DEH
IV	660BH	3A72H
V	7845H	D976H

Таблиця 5.6 - Результат помістити

Варіант	Внутрішнє ОЗП	Порти
I	6DH	P0, P3
II	27H	P1, P0
III	4FH	P2, P1
IV	2EH	P3, P0
V	2AH	P1, P2

5.2.4 Розділити два 16-розрядних числа X и Y та розмістити результат згідно з варіантом табл. 5.7-5.8.

Таблиця 5.6 - Розділити два 16-розрядних числа X и Y

Варіант	X	Y
I	3A06H	2005H
II	101FH	01A5H
III	E02AH	A10DH
IV	A101H	040AH
V	F486H	DABAH

Таблиця 5.7 - Результат помістити

Варіант	Внутрішнє ОЗП	Порт
I	20H	P0, P1
II	4BH	P3, P0
III	5FH	P0, P2
IV	28H	P1, P3
V	56H	P2, P3

ЛИТЕРАТУРА

1. Каспер Э. Программирование на языке Ассемблера для микроконтроллеров семейства i8051 / Э. Каспер – М.: Горячая линия-Телеком, 2004. – 192 с.
2. Магда Ю. С. Микроконтроллеры серии 8051: практический подход / Ю. С. Магда — М.: ДМК Пресс, 2008. — 228 с.
3. Гуртовцев А.Л. Программы для микропроцессоров: Справ. пособие / А.Л. Гуртовцев, С.В. Гудыменко – Мн.: Выш. шк., 1989. – 352 с.
4. Сташин В. В. Проектирование цифровых устройств на однокристалльных микроконтроллерах / В. В. Сташин – М.: «Энергоатомиздат», 1990. – 224 с.
5. Башков Е. А. Аппаратное и программное обеспечение зарубежных микроЭВМ: учебное пособие / Е. А. Башков – К.: «Высшая шк.», 1990. – 207 с.
6. Однокристалльные микроЭВМ: Справочник / А. В. Боборыкин, Г. П. Липовецкий, Г. В. Литвинский, О. Н Оксинь – М.: «МИКАП», 1994. – 400 с.
7. Щелкунов Н. Н. Микропроцессорные средства и системы / Н. Н. Щелкунов, А. П. Дианов – М.: «Радио и связь», 1989 – 288 с.
8. Максимов А. PROTEUS VS Система виртуального моделирования схем / Режим доступа: <http://www.twirpx.com/file/483689/>

Додаток А

Арифметичні операції

Цю групу утворюють 24 команди (табл. П1.2), які виконують операції додавання, десяткової корекції, інкремента/декремента байтів. Є команди віднімання, множення та ділення байтів.

Команди ADD та ADDC допускають складення(додавання) акумулятора з великою кількістю операндів. Аналогічно командам ADDC існують чотири команди SUBB, що дозволяє достатньо просто виконувати віднімання байтів та багатобайтних двійкових чисел.

В мікроконтролері реалізується розширений список(перелік) команд інкремента/декремента байтів, команда інкремента 16-бітного регістра-вказівника (покажчика) даних.

Приклади використання команд арифметичних операцій

Приклад 2. Додавання. Додати два двійкові багатобайтні числа. Доданки розміщуються в резидентній пам'яті даних, починаючи з молодшого байту. Початкові адреса доданків задані в R0 та R1, формат доданків у байтах – в R2:

```
CLR C           ; скидання переносу
LOOP:  MOV A, @R0      ; завантаження в А поточного байту
                        ; першого доданку
        ADDC A, @R1     ; додавання байтів з урахуванням переносу
        MOV @R0, A      ; розміщення байту результату
        INC R0          ; просування вказівників (покажчиків)
        INC R1
        DJNZ R2, LOOP   ; цикл, якщо не всі байти підсумовані
```

При додаванні чисел без знаку, на переповнення вкаже прапор C, а у випадку додавання чисел із знаком – прапор OV.

Доповніть програму додавання командами, які забезпечують її тестування, створіть контрольний приклад і виконайте налагодження в ProView. Визначте час обчислення в залежності від формату вихідних чисел.

Приклад 3. Множення. Команда MUL обчислює добуток двох цілих без знакових чисел, які зберігаються в регістрах А і В. Молодша

частина добутку розміщається в А, а старша в регістрі-розширювачі В. Якщо вміст В виявляється рівним нулю, то прапор OV скидається, інакше – встановлюється. Прапор переносу завжди скидається. Наприклад, якщо акумулятор містив число 200 (0C8H), а розширювач 160 (0A0H), то в результаті виконання команди MUL AB отримаємо добуток 32000 (7D00H). Акумулятор буде містити нуль, а розширювач – 7DH, прапор OV буде встановлений, а прапор C – скинутий.

Нехай потрібно помножити ціле двійкове число довільного формату на константу 73. Вихідне число розміщається в резидентній пам'яті даних, адреса молодшого байту знаходиться в регістрі R0. Формат числа в байтах зберігається в R1:

MOV A, #0	; скидання акумулятора
LOOP: ADD A, @R0	; завантаження множеного
MOV B, #73	; завантаження множника
MUL AB	; множення
MOV @R0, A	; запис молодшого байту часткового
	; добутку
INC R0	; приріст адреси
MOV A, B	; пересилання старшого байту
	; часткового добутку в акумулятор
XCH A, @R0	; попереднє формування чергового
	; байту добутку
DJNZ R1, LOOP	; цикл, якщо не всі байти вихідного
	; числа помножені на константу

Отриманий добуток розміщається на місці вихідного числа і займає в пам'яті на один байт більше.

Розберіться в алгоритмі множення. Доповніть програму командами, які забезпечують її тестування, складіть контрольний приклад і виконайте налагодження в ProView.

Приклад 4. Ділення. Команда DIV виконує ділення вмісту акумулятора на вміст регістра – розширювача. Після ділення акумулятор містить цілу частину частки, а розширювач – залишок. Прапори C і OV скидаються. При діленні на нуль встановлюється прапор переповнення, а частка залишається невизначеною. Команда ділення може бути використана для швидкого перетворення двійкових чисел в десяткові двійково – кодовані (BCD - числа).

У якості приклада розглянемо програму , яка переводить двійкове число, що міститься в акумуляторі, в BCD – код. При такому перетворенні може вийти трьохрозрядне BCD – число. Старша цифра (число сотень) буде розміщена в регістрі R0, а дві молодші в акумуляторі:

MOV B, #100	; завантаження 100 для обчислення кількості сотень
DIV AB	; акумулятор містить число сотень (старшу цифру)
MOV R0, A	; пересилання в R0 старшої цифри
XCH A, B	; пересилання залишку вихідного числа в акумулятор
MOV B, #10	; завантаження 10 для обчислення кількості десятків
DIV AB	; A містить число десятків, B – число одиниць
SWAP A	; розміщення числа десятків в старшому тетраді A
ADD A, B	; підсумовування залишку (числа одиниць), тепер
	; акумулятор містить дві молодші цифри

Розберіться в алгоритмі перевodu. Доповніть програму командами, які забезпечують її тестування, складіть контрольний приклад і виконайте налагодження в ProView. Визначте час обчислення.