

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з дисципліни
" Комп'ютерна схемотехніка " для студентів фаху 8.091501
«Комп'ютерні системи та мережі» та 8.091503 «Спеціалізовані
комп'ютерні системи» усіх форм навчання

2016

Методичні вказівки до виконання курсової роботи з дисципліни "Комп'ютерна схемотехніка" для студентів фаху 8.091501 «Комп'ютерні системи та мережі» та 8.091503 „Спеціалізовані комп'ютерні системи” усіх форм навчання / Укл. С.О. Сгадов, С.С. Грушко– Запоріжжя: ЗНТУ, 2016. – 20 с.

Укладачі: С.О. Сгадов , ст. викладач,
С.С. Грушко, асистент,

Рецензент: Р.К. Кудерметов, к.т.н., доцент

Відповідальний за випуск: С.О. Сгадов, ст. викладач

Затверджено
на засіданні кафедри
«Комп'ютерні системи та мережі»
Протокол №3 від 18.10.2016

Рекомендовано до видання НМК
ФКНТ
Протокол № 2
від “20” 10 2016.

ЗМІСТ

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	4
1.1 Мета і завдання до курсової роботи	4
1.2 Тематика курсових робіт	4
1.3 Зміст і обсяг пояснювальної записки	4
1.4 Календарний план виконання курсової роботи.....	5
1.5 Алгоритми.....	5
1.6 Оформлення графічного матеріалу.....	7
1.7 Захист курсового проекту.....	7
2 СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ ЩОДО АРХІТЕКТУРИ МІКРОПРОЦЕСОРІВ	8
3 АДРЕСАЦІЯ ТА СИСТЕМА КОМАНД МК-51	13
3.1 Система команд МК-51.....	13
3.2 Адреси бітів регістрів спеціальних функцій та адресація пам'яті.....	19
ЛІТЕРАТУРА	22

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Мета і завдання до курсової роботи

Курсовий проект є одним з етапів навчання студентів у ВНЗ і має своєю метою:

- систематизацію, закріплення та розширення теоретичних і практичних знань по спеціальності і застосування цих знань при виконанні конкретних задач;
- розвиток навичок ведення самостійної роботи;
- освоєння методів обґрунтування прийнятих рішень з урахуванням технічних вимог при розробці конкретної мікропроцесорної системи.

1.2 Тематика курсових робіт

Тема курсового проекту повинна відповідати дисципліні спеціальності.

Курсовий проект повинен містити:

- обґрунтування обраної системи та ПЗ;
- аналіз поставленої задачі та можливих варіантів побудови системи і ПЗ, вибір критеріїв, що визначають нові якості проектованої системи та ПЗ;
- синтез структури, ПЗ та окремих елементів системи на основі засад системного підходу до проектування.

1.3 Зміст і обсяг пояснювальної записки

Пояснювальна записка є основним документом, що входить у курсовий проект. Текст записки повинен містити оброблений та систематизований матеріал, що має безпосереднє відношення до теми курсового проекту.

Загальними вимогами до пояснювальної записки є:

- логічна послідовність викладення матеріалу у відповідності до поставленого завдання;
- стислість та чіткість формулювань, конкретність викладення результатів аналізу, розрахунків або дослідів;
- єдність термінів і позначень у межах проекту та їх відповідності з прийнятим у науково-технічній літературі та в ЄСПД.

Структура пояснювальної записки оформлюється у відповідності з діючим стандартом підприємства (ЗНТУ) і повинна мати у зазначеній послідовності наступні елементи:

- титульний лист;

- завдання на проектування;
- реферат;
- зміст;
- перелік скорочень та спеціальних термінів із їх визначенням (у випадку необхідності);
- основну частину;
- список використаних джерел;
- додатки.

Обсяг записки повинен становити 30... 43 стор.

Зразковий розподіл обсягу пояснювальної записки за основними розділами:

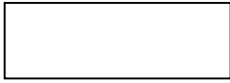
- титульний лист – 1стор.;
- завдання на курсовий проект – 1-2 стор.;
- реферат – 1 стор.;
- зміст – 1 стор.;
- вступ – 1-2 стор.;
- огляд і аналіз використаних джерел – 2-3 стор.;
- обґрунтування та розробку алгоритму, функціональної або структурної схеми системи, що розробляється – 5-7 стор.;
- структури розроблених алгоритмів і тексти програм з їх описом 10-15 стор.;
- розрахунок проектного приладу і результати роботи програм з аналізом та оцінкою похибок – 5-8 стор.;
- висновки – 1 стор.;
- список використаних джерел – 1-2 стор.;
- додатки – 1-2 стор.

1.4 Календарний план виконання курсової роботи

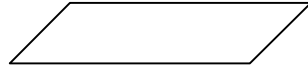
- Аналіз технічного завдання – 3 тижня;
- обґрунтування та розрахунок електричної схеми – 3 тижня;
- обґрунтування та розробка алгоритму – 1.5 тиждень;
- розробка та налагодження програми – 2 тиждень;
- оформлення пояснювальної записки -1 тиждень;
- оформлення графічного матеріалу – 1,5 тижня.

1.5 Алгоритми

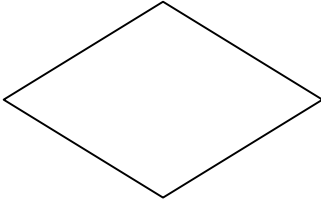
Алгоритми програм будуються у відповідності з ДЕСТ 19.003-80 "Єдина Система Програмної Документації. Схеми алгоритмів і програм. Позначення умовні графічні".



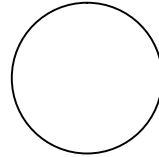
а)



г)



б)



д)



в)



е)

Рисунок 1.1 - Графічні позначення елементів алгоритму програм

Деякі символи, що найбільш використовуються на практиці, показані на рис. 1.1, де

а - процес виконання операції або групи операцій, у наслідку яких змінюється значення, форма представлення або розміщення даних;

б - рішення - вибір напрямку виконання алгоритму або програми у залежності від деяких змінних умов;

в - перевизначений процес - використання раніш створених та окремо описаних алгоритмів чи програм;

г - введення/виведення - перетворення даних у форму, придатну для обробки (введення) або відображення результатів обробки (виведення);

д - з'єднувач - вказівка зв'язку між перерваними лініями потоку, що зв'язують символи;

е - пуск/ зупин - початок, кінець, переривання процесу обробки даних або виконання програми.

1.6 Оформлення графічного матеріалу

При виконанні графічної частини курсового проекту необхідно додержувати норми і вимоги, встановлені стандартом підприємства (ЗНТУ).

Графічна частина курсового проекту виконується на 2-3 аркушах креслярського білого паперу формату А1 (594 x 841 мм). Допускається виконання графіків чорною тушшю.

У графічну частину проекту можуть входити:

- алгоритм програми;
- схема електрична функціональна;
- схеми електричні принципів каскадів, що розраховуються, і (або) усього приладу в цілому;
- вигляд різних характеристик приладу, розрахунок яких надається у пояснювальній записці;
- результати експериментальних досліджень.
- вся інформація, яка виноситься на демонстраційні листи, повинна міститись також у пояснювальній записці.

1.7 Захист курсового проекту

Допущений до захисту курсовий проект захищається комісії з двох викладачів.

Захист починається доповіддю студента, тривалістю до 7 хвилин. За цей час він повинен сформулювати поставлену задачу, показати актуальність теми, що опрацьовується, у стислій формі викласти основний зміст проекту, обґрунтувати прийняті рішення.

Результати захисту курсового проекту оцінюються по п'ятибальній системі і оголошуються у той же день. При визначенні оцінки проекту зважаються рівень теоретичної та практичної підготовки студента.

У тих випадках, коли захист курсового проекту визнається незадовільним, комісія встановлює, чи може студент подати до повторного захисту той самий проект із доробкою, що визначається комісією, або ж зобов'язаний розробити нову тему, що встановлюється кафедрою.

2 СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ ЩОДО АРХІТЕКТУРИ МІКРОПРОЦЕСОРІВ

Розглянемо особливості організації процесу обробки інформації у цифрових приладах.

Задача цифрового автомату, що виконує певні дії над бінарними сигналами, полягає у виборі елементів та засобів їх з'єднання, що забезпечують задане функціональне перетворення. Ці задачі вирішують за допомогою алгебри логіки.

Прилади, що формують функції алгебри логіки, називають цифровими і класифікують по різним ознакам. За схемним рішенням і характером зв'язків між вхідними та вихідними змінними з урахуванням їх змін по тактам роботи розрізняють два типи цифрових приладів - комбінаційні і цифрові. У комбінаційних цифрових приладах сукупність сигналів на виходах у кожний конкретний момент часу повністю визначається вхідними сигналами, діючими у цей момент на його входах.

Цифрові прилади послідовного типу суттєво відрізняються від комбінаційних насамперед наявністю пам'яті. Їхні вхідні сигнали є функцією не лише вхідних сигналів, але й внутрішнього стану, в якому прилад знаходився до надходження вхідних сигналів.

На основі цифрового приладу послідовного типу може бути спроектований прилад, що у залежності від послідовності вхідних сигналів буде виконувати один з багатьох алгоритмів. Ці вхідні сигнали можуть розміщуватися та послідовно вилучатися із зовнішнього блоку регістрів, що називається керуючою пам'яттю. Такий прилад може бути названий приладом із програмувальною логікою. До таких приладів відносяться і мікропроцесори.

Архітектура мікропроцесора являє собою логічну організацію, що визначає можливості апаратної або програмної реалізації функцій, необхідних для побудови мікроЕОМ.

Характеристики мікропроцесорів:

- розрядність адреси і даних;
- тип корпусу;
- кількість джерел живлення;
- потужність розсіювання;
- температурний діапазон;

- можливість розширення розрядності;
- час циклу виконання команд;
- рівні сигналів;
- завадостійкість;
- навантажувальна спроможність об'єднання сигналів на виходах;
- надійність.

По числу ВІС у мікропроцесорному комплекті розрізняють однокристальні, багатокристальні, багатокристальні секційні МП.

Однокристальні МП одержуються при реалізації усіх апаратних засобів процесора у вигляді однієї ВІС. По мірі збільшення ступеню інтеграції елементів в кристалі та числа виводів корпусу параметри однокристалевих МП покращуються.

Багатокристальні МП одержуються при розбитті його логічної структури на функціонально закінчені частини, що реалізуються у вигляді ВІС.

Багатокристалеві секційні МП одержуються у тому випадку, коли у вигляді ВІС реалізуються частини логічної структури процесора.

За призначенням розрізняють універсальні і спеціалізовані МП.

За виглядом вхідних даних, що обробляються, розрізняють цифрові та аналогові МП.

За характером часової організації роботи розрізняють синхронні та асинхронні МП.

Синхронні МП - це МП в яких початок і кінець виконання операцій задаються приладом управління.

Асинхронні МП дозволяють початок кожної наступної операції визначити за сигналом фактичного закінчення виконання попередньої операції. До складу асинхронних приладів вводять електронні кола, що забезпечують автономне функціонування приладів.

За кількістю програм, що виконуються розрізняють одно- та багатопрограмні МП.

Типова структура мікропроцесора наведена на рис. 3.1. Мікропроцесор складається з трьох основних блоків: арифметично-логічного приладу (АЛП), блока внутрішніх регістрів та приладу управління. Для передачі даних між цими блоками використовується внутрішня шина даних. Арифметично-логічний прилад виконує одну з

головних функцій мікропроцесора - обробку даних. Перелік функцій АЛП залежить від типу мікропроцесора. Деякі АЛП спроможні виконувати безліч різних операцій, у інших набір операцій обмежений. Функції АЛП визначають архітектуру мікропроцесора в цілому. Операції, що виконуються АЛП, у більшості мікропроцесорів наступні: додавання, віднімання, І, АБО, виключаюче АБО, інверсія, зсув праворуч, зсув ліворуч, приріст позитивний та негативний.

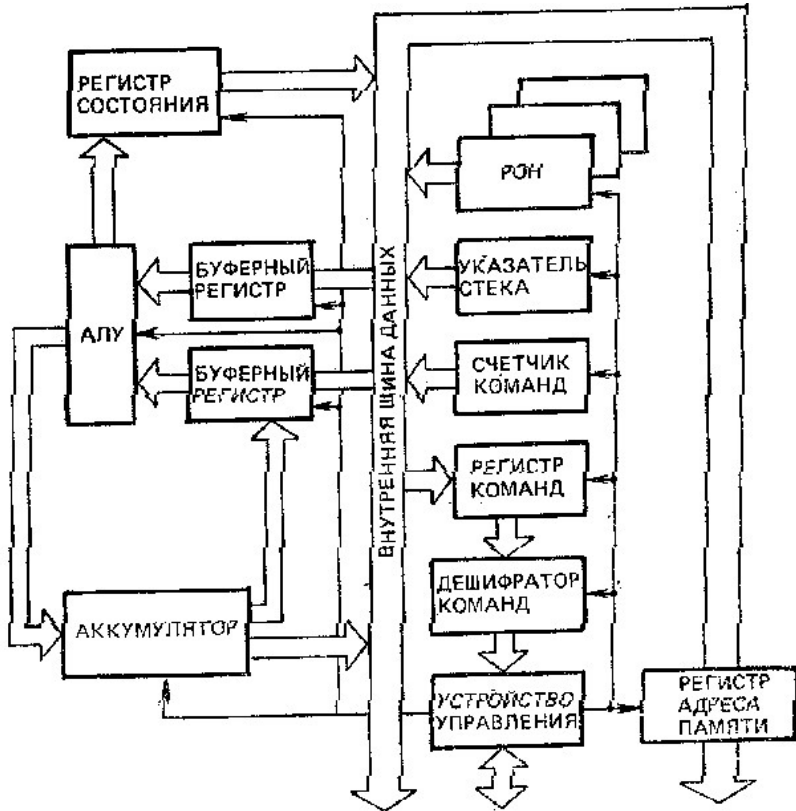


Рисунок 2.1 - Типова структурна схема МП

Важлива складова мікропроцесора - реєстри. Кожний реєстр мікропроцесора можна використовувати для тимчасового зберігання одного слова даних. Деякі реєстри мають спеціальне призначення, інші - багатоцільове. Останні зуться реєстрами загального призначення (РЗП) і можуть використовуватися програмістом на його розсуд. Кількість і призначення реєстрів у мікропроцесорі залежать від його архітектури.

Розглянемо призначення основних реєстрів, що є майже в усіх мікропроцесорах.

Акумулятор - це головний реєстр мікропроцесора при різноманітних маніпуляціях з даними. Більшість арифметичних та логічних операцій здійснюється шляхом використання АЛП та акумулятора. Будь-яка з таких операцій над двома словами даних (операндами) припускає розміщення одного з них у акумуляторі, а іншого у пам'яті або якомусь реєстрі. Так, при додаванні двох слів, що зуться умовно А і В розташованих у акумуляторі та пам'яті відповідно, результуюча сума С завантажується до акумулятору, замішуючи слово А. Результат виконання операції АЛП теж зазвичай розміщується у акумуляторі, вміст якого при цьому втрачається.

Лічильник команд - це один з найбільш важливих реєстрів МП. Як відомо, програма - це послідовність команд, які зберігаються у пам'яті мікроЕОМ і призначені для того, щоб інструктувати машину, як вирішувати поставлену задачу. Для коректного її виконання команди повинні надходити у строго певному порядку. Лічильник команд забезпечує формування адреси чергової команди, записаної у пам'яті.

Реєстр команд - містить команду у процесі її дешифрування та виконання.

Реєстр адреси пам'яті - при кожному зверненні до пам'яті мікроЕОМ вказує адресу області пам'яті, що належить використанню мікропроцесором. Реєстр адреси пам'яті містить бінарне число - адресу області пам'яті. Вихід цього реєстру називається адресною шиною і використовується для вибору області пам'яті або порту вводу-виводу.

Буферний реєстр - призначений для тимчасового зберігання даних.

Реєстр стану - призначений для зберігання результатів деяких перевірок, що здійснюються у процесі виконання програми.

Регістри загального призначення - реєстри, що використовуються у якості запам'ятовуючих приладів.

Показчик стеку - набір реєстрів МП або комірок ОП, звідки дані або адреси обираються "згори" по принципу: першим надійшов – останнім вийшов. При запису у стек чергового слова всі раніш записані слова зсуваються на один реєстр вниз. При вибірці слова зі стека, слова, що залишилися, переміщуються на один реєстр вгору.

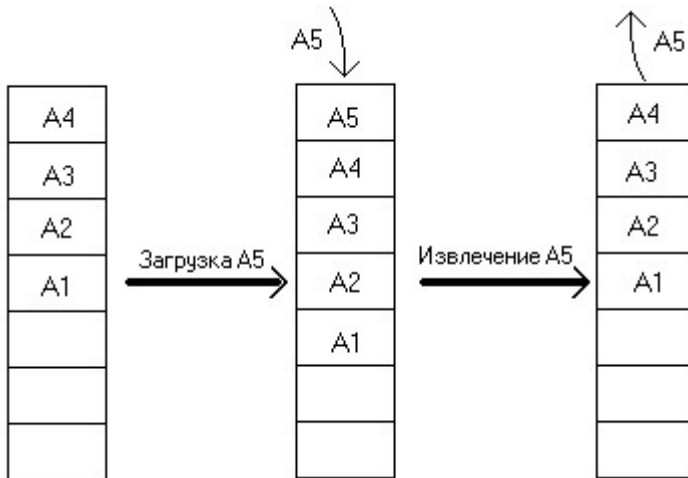


Рисунок 2.2 - Принцип роботи стека

- універсальний синхронно - асинхронний програмувальний прийомопередавач KP580BB51;
- програмувальний контролер електронно-променевої трубки KP580BG75;
- програмувальний контролер клавіатури KP580BB79;
- системний контролер KP580BH28.

3 АДРЕСАЦІЯ ТА СИСТЕМА КОМАНД МК-51

3.1 Система команд МК-51

Система команд ОЕОМ надає великі можливості обробки даних, забезпечує реалізацію логічних, арифметичних операцій, а також управління у режимі реального часу. Реалізована побітова, потетрадна (4 біта), побайтова (8 біт) і 16-розрядна обробка даних.

Команда ADD A, <байт джерело>

Ця команда складає вміст акумулятора А з вмістом байта-джерела, залишаючи результат у акумуляторі. При появі перенесення з розрядів 7 і 3, встановлюються прапори перенесення (C) та додаткового перенесення (AC) відповідно, у протилежному випадку ці прапори скидаються. При додаванні цілих чисел без знаку прапор перенесення (C) вказує на виконання переповнення. Прапор переповнення (OV) встановлюється, якщо є перенесення з біта 6 і немає перенесення з біта 7, або є перенесення з біта 7 і немає - з біта 6, у протилежному випадку прапор (OV) скидається. При додаванні цілих чисел зі знаком прапор (OV) вказує на негативну величину, одержану при додаванні двох позитивних операндів або на позитивну суму для двох негативних операндів.

Для команди додавання дозволені наступні режими адресації байта джерела:

- регістровий;
- побічно-регістровий;
- прямий;
- безпосередній.

Команда CLR A

Команда "скидання акумулятора" скидає (обнулює) вміст акумулятора А. На прапори команда не впливає.

Команда CLR <bit>

Команда "скидання" скидає вказаний біт в нуль. Ця команда працює з прапором перенесення (C) чи будь-яким бітом з прямою адресацією.

Команда CPL A

Команда "інверсія акумулятора" кожний біт акумулятора інвертує (змінює на протилежний). Біти, що містять "одиниці", після

цієї команди будуть містити "нулі", і навпаки. На прапори ця операція не впливає.

Команда CPL <bit>

Команда "інверсія біту" інвертує (змінює на протилежне значення) означений біт. Біт, що був "одиницею", змінюється у "нуль" і навпаки. Команда CPL може працювати з прапором перенесення або з будь-яким бітом, що прямо адресуються. На інші прапори команда не впливає.

Примітка. Якщо ця команда використовується для зміни інформації на виході порту, значення, що використовується як вихідні дані, зчитується з "защипки" порту, а не із виводів ВІС.

Команда DEC <байт>

Команда "декремент" робить віднімання '1' з означеного операнда. Початкове значення 00H перейде у 0FFH.

Команда DEC не впливає на прапори. Цією командою допускається чотири режими адресації операнда:

- до акумулятора;
- регістровий;
- прямий;
- побічно-регістровий.

Примітка. Якщо ця команда використовується для зміни інформації на виході порту, значення, що використовується як вихідні дані, зчитується з "заклучки" порту, а не з виводів ВІС.

Команда DIV AB

Команда "ділення" ділить 8-бітове ціле без знаку з акумулятора А на 8-бітове ціле без знаку у регістрі В. Акумулятору привласнюється ціла частина частки (старші розряди), а регістру В - залишок. Прапори перенесення (C) та переповнення (OV) скидаються. Якщо (A) < (B), то прапор додаткового перенесення (AC) не скидається. Прапор перенесення скидається в будь-якому випадку.

Примітка. Якщо В містить 00, то після команди DIV вміст акумулятора А та регістру В будуть невизначені. Прапор перенесення скидається, а прапор переповнення встановлюється у '1'.

Команда INC <байт>

Команда "інкремент" виконує додання '1' до означеної змінної і не впливає на прапори. Початкове значення 0FFH перейде у 00H. Ця команда припускає 4 режими адресації:

- до акумулятора;

- реєстровий;
- прямий;
- непрямо-реєстровий.

Примітка. При використанні команди INC для зміни вмісту порту, величина, що використовується як операнд, зчитується з "заключки" порту, а не з виводів BIC.

Команда INC DPTR

Команда "інкремент показчика даних" виконує інкремент (додання '1') вмісту 16-бітового показчика даних (DPTR). Додання '1' здійснюється до 16 бітів, причому переповнення молодшого байта показчика даних (DPL) з FFH у 00H приводить до інкременту старшого байта показчика даних (DPH). На прапори ця команда не впливає.

Команда JB <bit>, <rel8>

Команда "перехід, якщо біт встановлений" виконує перехід за адресою галуження, якщо означений біт рівний '1', у протилежному випадку виконується наступна команда. Адреса галуження обчислюється за допомогою додання відносного зміщення зі знаком у третьому байті команди (rel8) до вмісту лічильника команд після додання до нього 3. Біт, що перевіряється, не змінюється. Ця команда на прапори не впливає.

Команда JMP @+DPTR

Команда "непрямий перехід" складає восьми бітовий вміст акумулятора без знаку з 16-бітовим показчиком даних (DPTR) і завантажує отриманий результат у лічильник команд, вміст якого є адресою для вибору наступної команди. 16-бітове додавання виконується по модулю 2^{16} , перенесення з молодших восьми бітів поширюється на старші біти програмного лічильника. Вміст акумулятора і показчика даних не змінюється.

Команда JNB <bit>, <rel8>

Команда "перехід, якщо біт не встановлений" виконує галуження за адресою, якщо означений біт рівний '0', у протилежному випадку виконується наступна команда. Адреса галуження обчислюється за допомогою додавання відносного зміщення зі знаком у третьому байті команди (rel8) і вмісту лічильника команд, після додання до нього 3. Біт, що перевіряється, не змінюється. Ця команда не впливає на прапори.

Команда JNZ <rel 8>

Команда "перехід, якщо вміст акумулятора не рівний нулю" виконує галуження за адресою, якщо хоча б один біт акумулятора рівний '1', у протилежному випадку виконується наступна команда. Адреса галуження обчислюється додаванням відносного зміщення зі знаком в другому байті команди (rel8) та вмісту лічильника команд (PC) після додання до нього 2. Вміст акумулятора не змінюється. Ця команда на прапорі не впливає.

Команда JZ <rel8>

Команда "перехід, якщо вміст акумулятора рівний '0'" виконує галуження за адресою, якщо всі біти акумулятора рівні '0', у протилежному випадку виконується наступна команда. Адреса галуження обчислюється додаванням відносного зміщення зі знаком в другому байті команди (rel8) та вмістом лічильника команд після додання до нього 2. Вміст акумулятора не змінюється. Ця команда на прапорі не впливає.

Команда MOV <байт призначення>, <байт джерела>

Команда "переслати змінну-байт" пересилає змінну-байт, означену у другому операнді у другі комірку, означену у першому операнді. Ця команда на прапорі та інші регістри не впливає. Вміст байта-джерела не змінюється. Ця команда припускає 15 комбінацій адресації байта-джерела і байта призначення.

Команда MOV <біт призначення>, <біт джерела>

Команда "переслати біти даних" бітову змінну, означену у другому байті, копіює у розряд, що вказаний у першому операнді. Одним з операндів повинен бути прапор перенесення (C), а другим може бути будь-який біт, до якого можлива пряма адресація.

Команда MOV DPTR, #data16

Команда "завантажити покажчик даних 16 - бітовою константою" завантажує покажчик даних DPTR 16-бітовою константою, означеною в другому та третьому байтах команди. Другий байт команди завантажується у старший байт покажчика даних (DPH), а третій байт - у молодший байт покажчика даних (DPL). Ця команда на прапорі не впливає і є єдиною командою, що водночас завантажує 16 біт даних.

Команда MOVC A, @A+ (<R16>)

<R16> --16-розрядний регістр.

Команда "переслати байт з пам'яті програм" завантажує акумулятор байтом коду або константою з пам'яті програми. Адреса

байта, що зчитується, обчислюється як сума 8-бітового вихідного вмісту акумулятора без знаку та вмісту 16-бітового регістру.

У якості 16-бітового регістру може бути:

- показчик даних DPTR,
- лічильник команд PC.

У випадку, коли використовується PC, він збільшується до адреси наступної команди перед тим, як його вміст складається з вмістом акумулятора. 16-бітове додавання виконується так, що перенесення з молодших восьми бітів може поширюватися через старші біти. Ця команда на прапори не впливає.

Команда MOVX <байт приймача>, <байт джерела>

Команда "переслати в зовнішню пам'ять (з зовнішньої пам'яті) даних" пересилає дані між акумулятором та байтом зовнішньої пам'яті даних. Існує два типи команд, що відрізняються тим, що забезпечують 8-бітову або 16-бітову непряму адресу до зовнішнього ОЗП даних.

У першому випадку вміст R0 або R1 у поточному банку регістрів забезпечує 8-бітова адреса, що мультиплексується з даними порту P0. Для розширення дешифрації вводу-виводу чи адресації невеликого масиву ОЗП достатньо восьми бітів адресації. Якщо застосовується ОЗП, трохи більше ніж 256 байт, то для фіксації старших бітів адреси можна використовувати будь-які інші виводи портів, що перемикаються командою, яка стоїть перед командою MOVX.

В іншому випадку, при виконанні команди MOVX показчик даних DPTR генерує 16-бітову адресу. Порт P2 виводить старші 8 бітів адреси (DPH), а порт P0 мультиплексує молодші 8 бітів адреси (DPL) з даними.

Ця форма є ефективною при доступі до великих масивів даних (до 64K), бо для установки портів виведення не вимагається додаткових команд.

Команда ORL <байт призначення>, <байт джерела>

Команда "логічне "АБО" для змінних-байтів" виконує операцію логічного "або" над бітами вказаних змінних, записуючи результат у байт призначення. Ця команда на прапори не впливає. Допускається 6 комбінацій режимів адресації:

- якщо байтом призначення є акумулятор:
 - 1) регістровий;
 - 2) прямий;

- 3) побічно-регістровий;
- 4) безпосередній ;
- якщо байтом призначення є пряма адреса:
 - 5) до акумулятору;
 - 6) до константи .

Примітка. Якщо команда використовується для роботи з портом, величина, що використовується у якості вихідних даних порту, зчитується з "заключки" порту, а не з виводів VIC.

Команда POP <direct>

Команда "читання зі стеку" зчитує вміст комірки, що адресується за допомогою покажчика стека, до комірки ОЗП, що прямо адресується, при цьому покажчик стека зменшується на одиницю. Ця команда не впливає на прапори і часто використовується для читання зі стека проміжних даних.

Команда PUSH <direct>

Команда "запис у стек" збільшує покажчик стека на одиницю і після цього вміст означеної змінної, що прямо адресується, копіюється в комірку внутрішнього ОЗП, що адресується за допомогою покажчика стека. На прапори ця команда не впливає і використовується для запису проміжних даних у стек.

Команда RET

Команда "повернення з підпрограми" послідовно вивантажує старший та молодший байти лічильника команд зі стека, зменшуючи покажчик стека на 2. Виконання основної програми звичайно триває за адресою команди, наступної за ACALL або LCALL. На прапори ця команда не впливає.

Команда SETB

Команда "встановити біт" встановлює означений біт у 1.
Адресується:

- до прапора перенесення (C);
- до біта з прямою адресацією.

Команда SWAP A

Команда "обмін тетрадами всередині акумулятора" здійснює обмін між молодшими чотирма та старшими чотирма бітами акумулятора (між старшою і молодшою тетрадами).

Ця команда може розглядатися також, як команда чотирибітового циклічного зсуву. На прапори ця команда не впливає.

Команда XCH A, <byte>

Команда "обмін вмісту акумулятора із змінною-байтом" здійснює обмін вмісту акумулятора із вмістом джерела, означеним у команді. Операнд джерела може використовувати наступні режими адресації:

- регістровий;
- прямий;
- непрямо-регістровий.

3.2 Адреси бітів регістрів спеціальних функцій та адресація пам'яті

Зовнішня пам'ять даних формується додатковими мікросхемами пам'яті, які підключаються до OEOM і може мати ємність до 64 Кбайт. Простори внутрішньої і зовнішньої пам'яті даних не перетинаються, так як доступ до них здійснюється за допомогою різних команд. Для роботи із зовнішньою пам'яттю даних існують спеціальні команди MOVX, що не впливають на внутрішню пам'ять даних OEOM.

Таким чином, у системі можуть водночас бути внутрішня пам'ять даних з адресами 00H - FFH і зовнішня пам'ять даних з адресами 0000H - FFFFH. Звертання до комірок зовнішньої пам'яті даних здійснюється лише з використанням побічної адресації по регістрам R0 та R1 активного банку регістрів внутрішнього ОЗП (команди типу MOV @Ri) або по регістру спеціальних функцій DPTR (команди типу MOV @DPTR). Відповідно у першому випадку буде формуватися 8-розрядна, а в другому випадку 16-розрядна адреси зовнішньої пам'яті даних.

При зверненнях до зовнішньої пам'яті даних адреса вводиться через порт P0 (молодший байт) і порт P2 (старший байт) OEOM. Обмін байтом даних (запис та читання) здійснюється через порт P0 OEOM, тобто порт P0 використовується як шина адреси/даних у режимі мультиплексування. Зчитування даних з зовнішньої пам'яті даних до OEOM робиться за допомогою вихідного сигналу OEOM RD, а запис даних з OEOM у зовнішню пам'ять даних за допомогою вихідного сигналу OEOM WR.

Кожний тип зовнішньої пам'яті (пам'ять програм, пам'ять даних) може бути додано незалежно від іншого і кожний використовує ті ж адреси і шини даних, але різні сигнали управління.

4 ПРИКЛАДИ ТЕМ

- 1 **Термореле.** Спроекувати функціональну електричну схему й програму для ПЗУ мікропроцесорного пристрою для виміру температури, який виробляє керуючий сигнал, по перевищенню заданої користувачем температури. Температура вимірюється за допомогою термодатчика з лінійною характеристикою з точністю до градуса й індицирується на двухразрядном семисегментном індикаторі.
- 2 **Контролер коробки передач.** Спроекувати функціональну електричну схему й програму для ПЗУ мікропроцесорного пристрою контролера коробки передач. Цей контролер для захисту двигуна від перегріву при перевищенні обертів і температури. При перевищенні кількості обертів і/або температури двигуна вище критичної виробляє сигнал для виконавчого механізму перемикавання передач у послідовному коді для підвищення номера передачі; при зниженні - перемикає на попередній щабель передач. Контролер повинен дочекатися сигналу від коробки передач про завершення перемикавання. (Перемикавання передач не може здійснюватися занадто часто, тому що двигуну потрібно час для переходу в режим, і крім того, не слід занадто зношувати механізм коробки передач).
- 3 **Сонар.** Спроекувати функціональну електричну схему й програму для ПЗУ мікропроцесорного пристрою для виміру відстані за допомогою звуку. Через деякі проміжки часу виробляє клацання й вимірює час приходу відбитого сигналу, виводить відстань у метрах на індикатор. Акустичними перешкодами зневажити. Указати діапазон вірогідно вимірюваних відстаней виходячи зі швидкості мікропроцесора.
- 4 **Аналоговий скремблер.** Спроекувати функціональну електричну схему й програму для ПЗУ мікропроцесорного пристрою, який повинний забезпечити захист звукового сигналу діапазоні до 1000 Гц. На вхід подається акустичний сигнал - на виході - зашифрований за допомогою генератора випадкових чисел аналоговий сигнал. Генератор випадкових чисел

запускається після синхропосылки, яка передається перед зашифрованими даними.

- 5 **Цифровий скремблер.** Спроекувати функціональну електричну схему й програму для ПЗУ мікропроцесорного пристрою, який повинний забезпечити захист звукового сигналу діапазоні до 1000 Гц. На вхід подається акустичний сигнал - на виході - зашифрований за допомогою генератора випадкових чисел послідовний цифровий сигнал. Генератор випадкових чисел запускається при посилці сигналу кадрової синхронізації. (Інтерфейс – послідовний (лінія передачі, лінія синхронізації біт, лінія кадрової синхронізації, земля).
- 6 **Індикатор кута повороту.** Спроекувати функціональну електричну схему й програму для ПЗУ мікропроцесорного пристрою мікропроцесорний пристрій для виміру й індикації з точністю до 0.1° кута повороту щодо початкового. Вимір здійснюється двома датчиками. Основний датчик, приєднаний до вала, видає сигнали кожний 1, а прецизійний, прикріплений до вторинного вала через зубчасту передачу 10:1, видає 10 сигналів на 1 сигнал основного.
- 7 **Індикатор радіоактивності..** Спроекувати функціональну електричну схему й програму для ПЗУ мікропроцесорного пристрою, що підраховує кількість розпадів (сигналів на лічильнику Гейгера) за 15 сек і виводить кількість розпадів у хвилину на індикатор. (Слід передбачити гальванічну розв'язку процесора з виходом лічильника Гейгера, який є високовольтним пристроєм)
- 8 **Захисний пристрій.** Спроекувати функціональну електричну схему й програму для ПЗУ мікропроцесорного пристрою, призначене для охорони приміщення. Через 1 хв вимірює природній фон Ік-Випромінювання. Якщо воно міняється, то видається звуковий сигнал тривоги.

ЛІТЕРАТУРА

1. Балашов Е.П. и др. Микро- и миниЭВМ. /Е.П. Балашов, В.Л. Григорьев, Г.А. Петров. Учебное пособие для ВУЗов. -Л.: Энергоатомиздат. Ленинградское отделение, 1984, -376 с.
2. Белов А.В. Самоучитель по микропроцессорной технике. – СПб.: Наука и Техника, 2003. - 224 с.
3. Варламов І. В., Касаткін І. Л. Мікропроцесори у побутовій техніці. - М.: Радіо зв'язок, 1989.-80 с.
4. Мячев А.А., Степанов В.Н. Персональные ЭВМ и микроЭВМ. Основы организации: Справочник / Под ред. А.А. Мячева. -М.: Радио и связь, 1991, -320 с.
5. Применение микроконтроллеров MCS-51 при проектировании электронных устройств. Новосибирск, 1996. – 256 с.
6. Самофалов К.Г. и др. Микропроцессоры. Библиотека инженера.- 2-е издание, перераб. и доп., К.:Техника, 1989, - 312 с.
7. Солонина А.И., Яковлев Л.А. Основы построения микропроцессорных систем. Учебн. пособие. – Л.:ЛЭИС, 1991.
8. Сташин В.В. и др. Проектирование цифровых устройств на однокристалльных микроконтроллерах. /В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева, -М.: Энергоатомиздат, 1990, -224 с.
9. Тули М. Справочное пособие по цифровой электронике: Пер с англ. -М.: Энергоатомиздат, 1990, 176с.
10. Угрюмов Е.П. Цифровая схемотехника. – СПб.: Питер, 2000.
11. Цифровые интегральные микросхемы: Справочник /П.П. Мальцев, Н.С. Долидзе, М.И. Критенко и др. -М.: Радио и связь, 1994, - 240 с.