# Міністерство освіти і науки України

**Кіровоградський національний технічний університет**

Кафедра програмування та захисту інформації

Методичні вказівки

до лабораторних робіт

Дисципліна : **Комп’ютерна схемотехніка**

**Лабораторна робота №6**

**Тема: Простий ввід/вивід даних, вивід даних. Робота з клавіатурою та індикацією**

**Кіровоград 2015**

**Лабораторна робота №6**

**Тема: Простий ввід/вивід даних, вивід даних/ Робота з клавіатурою та індикацією**

**Мета роботи:** Вивчення пристроїв вводу -виводу

**Теоретичні відомості**

У мікропроцесорної системі слово «пам'ять» має цілком певне значення, а саме воно асоціюється з деякою схемою йди безліччю схем, з якими можна легко зв'язати функції читання н записи. Інакше йде справа з поняттям «пристрій виводу», воно визначено не досить чітко. Пристроєм висновку може бути й порядкове друкувальний пристрій, і табло на свстоізлучающіх діодах, і електронно-променева трубка (ЕПТ). накопичувач, і гнучкий диск і будь-яке інше пристрій мікропроцесорної системи. В інтересах подальшого викладу визначимо пристрій виводу наступним чином:

«Пристрій виведення - це будь-який пристрій, окрім пам'яті, що сприймає дані, що передаються від мікропроцесора». Такий підхід до визначення пристрою виводу дозволяє представляти його у вигляді пристрою, здатного приймати від 1 до 8 біт в паралельному режимі. Причому отримання даних осу-шествляется в тому випадку, якщо на шині управління активізована ЛІНІЯ управління записом на пристрій виводу.

Функціонування пристрої виведення, так само як і функціонування пам'яті, можна розглянути з позиції взаємодії електричних сигналів. Відмінність полягає лише в тому, що пристрій виводу здійснює обробку сигналу управління, а пам'ять системи-ні.

Це справедливо для багатьох мікропроцесорів. Системи, в яких реалізується подібна взаємодія, називають системами з організацією виведення за аналогією зі зверненням до пам'яті.

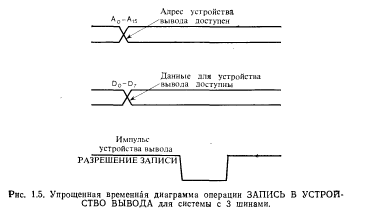
Кожному пристрою виводу приписана своя адреса. подібна адресі пам'яті системи. На рис. 1.5 показана в спрощеному вигляді тимчасова діаграма запису в пристрій виводу. З рис. 1.5 важливо зрозуміти, що адреса пристрою виводу і дані які надходять на відповідні шини до того, як шиною управління системи буде виданий імпульс дозволу виводу. Операція запису в пристрій виведення виконується шляхом peaлізаціі наступніх послідовностей дій:

1. Забезпечення стабільних рівнів сигналів А0-А15, відповідних адресі пристрою виводу. У деяких системах для адресації пристрої виводу використовуються не всі 16 адресних ліній.

2. Забезпечення на шині даних D0-D7; стабільних рівнів сигналів, відповідних даних, які повинні бути записані в пристрій виводу.

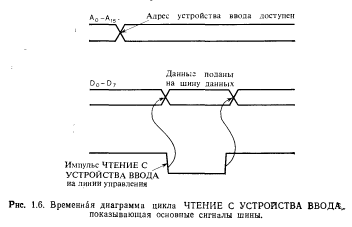
3. Після виконання кроків 1 і 2 подача на шину управління сигналу дозволу запису в пристрій виводу.

Розглянута вище операція запису даних в пристрій виведення виявляє функцію кожної шини системи. При реалізації цієї операції всі три шини функціонують у взаємодії один з одним. І використовуючи їх, система встановлює надійний зв'язок між мікропроцесором і пристроєм виводу.

****

Читання даних з пристрою введення

Як пристрій введення мікропроцесорної системи може розглядатися будь-яке джерело, крім пам'яті системи, здатний передавати дані мікропроцесору. Дані з пристрою введення пересилаються в мікропроцесор за запитом останнього. Як приклад пристрою введення можна назвати клавішний пульт терміналу з ЕПТ. Дані з пульта надходять в мікропроцесор по його запитам.

****

У деяких системах мікропроцесор не робить різниці між читанням з пристрою введення і з пам'яті системи. Подібне зауваження раніше було зроблено щодо влаштування-виводу. Якщо мікропроцесорна система не розрізняє читання даних з пам'яті і з пристрою введення, то говорять, що в системі використовується введення даних по аналогії із зверненням до пам'яті. Архітектура системи подібного типу згадувалася у зв'язку із записом даних на пристрій виводу.

На рис. 1.6 представлена ​​тимчасова діаграма, що показує взаємозв'язок між трьома шинами. З цього малюнка видно, Що адресу пристрої введення надходить на адресну шину до того, як на шину управління подається сигнал управління читанням з пристрою вводу. Протягом проміжку часу, поки на шині управління зберігається цей сигнал, дані з пристрою введення передаються на шину даних.

Порівнюючи розглянуті до сих пір тимчасові діаграми, можна помітити, що синхронізація процесів при читанні даних з пам'яті і з пристрою введення схожа. Те ж саме відноситься і до синхронізації процесів при запису даних в пам'ять і на пристрій виводу. За допомогою більшості мікропроцесорів можна будувати системи, в яких нема різниці між записом даних у пам'ять і на пристрій виводу або читанням даних з пам'яті і з пристрою вводу.

Введення інформації з кнопок і клавіатур

Загальні відомості

Введення інформації з кнопок і клавіатури, підключених до мікро-ЕОМ зазвичай пов'язаний з вирішенням наступних специфічних проблем:

1. захистом від брязкоту контактів кнопок або клавіш;

2. ідентифікацією натиснутою клавіші;

3. забезпеченням потрібного порядку спрацьовування клавіш (при натисненні або при відпуску).

Захист від брязкоту може бути забезпечена або на апаратній, або на програмному рівні.

Для ідентифікації клавіш звичайно потрібно комбінація певних апаратних і програмних засобів.

Потрібний порядок спрацьовування зазвичай забезпечується відповідним побудовою програми підтримки роботи клавіатури.

Ідентифікація натиснутою клавіші

Проблема ідентифікації натиснутою клавіші зазвичай вирішується в мікро-ЕОМ в два етапи. На першому етапі забезпечується генерація клавіатурою при натисканні кожної з клавіш унікального двійкового вода. Цей етап забезпечується або тільки апаратними засобами, або комбінацією апаратних і програмних засобів. Другий етап завжди забезпечується програмними засобами. На цьому етапі код натиснутої клавіші вводиться в мікро-ЕОМ і порівнюється із заздалегідь занесеної в пам'ять таблицею допустимих кодів. Результатом такого порівняння і є ідентифікація, впізнання натиснутою клавіші.

При малій кількості клавіш перший етап ідентифікації вирішується чисто апаратно. Для прийому сигналу з кожної з клавіш тут виділяють окремі розряди, біти в порте прямого введення. Приклад такого рішення ілюструє схема клавіатури, представлена на ріс.4.32. Комбінація натиснутих клавіш S0 ... S7 задає тут унікальний код натиснутої клавіші KNK1, який далі вводиться в мікро-ЕОМ через порт прямого введення PIKNK1. Якщо кількість клавіш більше числа розрядів порту введення, можна включити між клавіатурою і портом шифратор з відповідною кількістю входів. При використанні шифратора, восьмирозрядний порт введення може вводити інформацію максимум від 255 клавіш.

Другий етап ідентифікації натиснутою клавіші забезпечується підпрограмою IDEN2, яка:

1. вводить в мікро-ЕОМ код натиснутої клавіші KNK1;

2. перевіряє, чи міститься цей код в таблиці допустимих кодів;

1. встановлює ознака недопустимого введення Z = 1, якщо ця перевірка не успішна. Некоректний ввід можливий, наприклад, при одночасному натисканні декількох клавіш;

2. Встановлює ознака керуючої клавіші С = 1, якщо натиснута управляюча клавіша;

3. перетворює код натиснутої клавіші KNK1 в іншій - KNK2, більш зручний для подальшого використання.



Рис.1.1.Підключення клавіатури до мікро-ЕОМ

Для цифрових клавіш, коди KNK2 повинні відповідати їх маркування. Для керуючих клавіш зручно використовувати в якості кодів KNK2 послідовність парних шістнадцятиричних чисел, що починається з 00Н. Код KNK2 виходить в підпрограмі з проміжного коду CKNK2. Молодша тетрада коду CKNK2 є код KNK2 натиснутою клавіші. Старша тетрада дорівнює 0H для цифрових і 8H для керуючих клавіш. Така структура коду CKNK2 дозволяє одночасно використовувати його як для формування KNK2, так і ознаки керуючої клавіші.

У випадку, якщо на рис.1.2 клавіші S0 ... S3 - цифрові, а S4 ... S7 - керуючі, підпрограма IDEN2 може мати вигляд, представлений на рис.1.2.

Основу підпрограми становить цикл, в якому послідовно встановлюються адреси всіх восьми рядків таблиць ТАВ1 (KNK1) і ТАВ2 (CKNK2) і йде порівняння коду з таблиці ТАВ1 з кодом натиснутою клавіші. Якщо введеного коду немає в таблиці ТАВ1, тобто мав місце некоректний введення, відбувається вихід з підпрограми до встановленого некоректного введення Z = 1. Ця ознака встановлюється командою DCR B при завершенні циклу. Якщо ж введений код пізнаний в одному з кодів ТАВ1, слід вихід з циклу до мітці М1. Реєстрова пара DE містить в цьому випадку адреса коду CKNK2 в таблиці ТАВ2. Далі цей код витягується з ТАВ2 командою LDAX D.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | ; ПОДПРОГРАММА IDEN2 |
| PIKNK1 | EQU | … | ; ПОРТ ВВОДА PIKNK1 |
| LTAB | EQU | 08H | ; ДЛИНА ТАБЛИЦЫ КОДОВ |
| MASK | EQU | 0FH | ; МАСКА МЛАДШЕЙ ТЕТРАДЫ |
| IDEN2: | IN | PIKNK | ; ВВЕСТИ KNK1 |
|  | MVI | B,LTAB | ; ЗАГРУЗИТЬ ДЛИНУ ТАБЛИЦЫ |
|  |  |  | ; В РЕГИСТР B |
|  | LXI | H, TAB1 | ; ЗАГРУЗ. НАЧ. АДРЕС ТАБЛИЦЫ |
|  |  |  | ; КОДОВ KNK1 В ПАРУ HL |
|  | LXI | D, TAB2 | ; ЗАГРУЗ. НАЧ. АДРЕС ТАБЛИЦЫ |
|  |  |  | ; КОДОВ СKNK2 В ПАРУ DE |
| M0: | CMP | M | ; KNK1 СОВПАДАЕТ СО СТРОКОЙ |
|  |  |  | ;ТАБЛИЦЫ TAB1? |
|  | JZ | M1 | ; ДА, ПЕРЕЙТИ К М1, ИНАЧЕ |
|  | INX | H | ; МОДИФИЦИРОВАТЬ АДРЕСА |
|  | INX | D | ; КОДОВ В ТАВ1 И ТАВ2 |
|  | DCR | B | ; ПРОСМОТРЕНА ВСЯ ТАВ1? |
|  | JNZ | M0 | ; НЕТ, ПОВТОРИТЬ, ИНАЧЕ |
|  | RET |  | ; ВОЗВРАТ С Z=1 |
| M1: | LDAX | D | ; ВЫБРАТЬ CKNK2 ИЗ ТАВ2 |
|  | ANI | MASK | ; ВЫДЕЛИТЬ KNK2 |
|  | MOV | B, A | ; СОХРАНИТЬ В РЕГИСТРЕ В |
|  | MVI | A, 01H | ; УСТАНОВИТЬ |
|  | ORA | A | ; Z=0 |
|  | LDAX | D | ; ВЫБРАТЬ CKNK2 ИЗ ТАВ2 |
|  | RLC |  | ; ПЕРЕНЕСТИ СТАРШИЙ БИТ CKNK2 |
|  |  |  | ; В БИТ С РЕГИСТРА F |
|  | MOV | A, B | ; ВОССТАНОВИТЬ KNK2 В |
|  |  |  | ; РЕГИСТРЕ А |
|  | RET |  | ; ВОЗВРАТ С Z=0 И С=1 / 0 |
| TAB1: | DB | 0FEH, 0FDH, 0FBH, 0F7H | ; ЦИФРОВЫЕ КЛАВИШИ |
|  | DB | 0EFH, 0DFH, 0BFH, 7FH | ; УПРАВЛЯЮЩИЕ КЛАВИШИ |
| TAB2: | DB | 00H, 01H, 02H, 03H | ; ЦИФРОВЫЕ КЛАВИШИ |
|  | DB | 80H, 82H, 84H, 86H | ; УПРАВЛЯЮЩИЕ КЛАВИШИ |

Рис.1.2.Текст підпрограми IDEN2

Особливості ідентифікації натиснутою клавіші в матричної клавіатурі

У мікро-ЕОМ часто використовують так звану матричну клавіатуру. Така клавіатура являє собою прямокутну дротову матрицю, в вузлах якої включені контакти клавіш. Принципова схема одного з реальних варіантів матричної клавіатури наведена на рис.1.3

Роботу клавіатури підтримують порт прямого введення POKWR і порт прямого введення PIKAR. Легко бачити, що до кожної з чотирьох активних ліній порту введення тут підключено по 6 контактів клавіатури. Зрозуміло, що ці контакти не можуть бути опитані мікро-ЕОМ одночасно. Тому в процесі ідентифікації натиснутою клавіші використовується процедура послідовного опитування вертикальних рядів клавіш. Цю процедуру, здійснювану спеціальної підпрограмою, часто називають скануванням клавіатури.

Порт POKWR призначений для вибору ряду клавіш, опитуваних в даний момент часу. В цей порт виводиться код вибору ряду KWR. Одиничний біт цього коду забезпечує активацію, вибір одного з вертикальних рядів клавіш, задаючи на вертикальному провіднику матриці рівень логічного нуля. В невибраних рядах вертикальні провідники мають рівень логічної одиниці. Тому замикання контактів клавіатури в невибраних рядах не змінює одиничного стану входів D6, D5, D4 і D2 порту PIKAR. Замикання ж контактів в обраному ряду призводить до появи рівня логічного нуля на одному або декількох входах цього порту. Таким чином, формується код активного ряду KAR, який може ввести в мікро-ЕОМ через порт PIKAR.

Послідовний вибір рядів клавіш мінливим в циклі кодом KWR і введення коду KAR відповідного ряду здійснює спеціальний блок підпрограми ідентифікації IDEN. Байтові коди KWR і KAR складають разом унікальний шестнадцатіразрядний код натиснутої клавіші, що використовується на другому етапі ідентифікації. Так, наприклад, якщо цей код дорівнює 0170H, то натиснута клавіша «пробіл».

Для розглянутого варіанта клавіатури коди KWR і KAR можна об'єднати в байтовий код натиснутої клавіші KNK, використовуючи співвідношення

KNK = KWR / 2 KAR х 2.

Це співвідношення може бути реалізовано фрагментом програми, наведеним на рис.1.5. Тут передбачається, що код KWR міститься в регістрі С.

Часто потрібно визначити, натиснута якась (неважливо яка) з клавіш клавіатури. В цьому випадку слід вибрати всі шість рядів клавіш одночасно, видавши в порт POKWR код KWR = 3FH. Порт PIKAR буде при цьому брати код KAR = 74H тільки в тому випадку, якщо не натиснута жодна з клавіш.

Процеси сканування матричної клавіатури і динамічного управління семісегментним дисплеєм мають багато спільного. Тому їх часто інтегрують, поєднують на апаратній (клавіатура і дисплей зазвичай мають загальний порт вибору ряду POKWR) і (або) на програмному рівні.

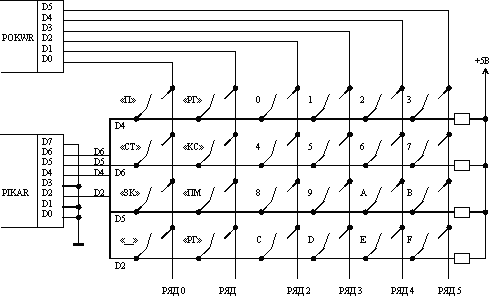


Рис.1.5.Принципова схема матричної клавіатури

|  |  |  |
| --- | --- | --- |
| … |  |  |
| IN | PIKAR | ; ВВЕСТИ КОД KAR |
| RLC |  | ; KAR x 2 |
| MOV | B,A | ; СОХРАНИТЬ KAR x 2 В РЕГ. В |
| MOV | A,C | ; ЗАГРУЗИТЬ KWR В РЕГ. А |
| RAR |  | ; KWR / 2 |
| ADD | B | ; KNK = KWR / 2 + KAR х 2 |
| … |  |  |

Рис.1.5.Об’єднання кодів KWR і KAR в код KNK

Програмна підтримка роботи клавіатури

Питання забезпечення потрібного порядку спрацьовування клавіш клавіатури, захисту від брязкоту її контактів та ідентифікації натиснутою клавіші зазвичай вирішує спеціальна підпрограма підтримки клавіатури KEY. Її типова блок-схема представлена на рис.1.6. Блоки 1 і 2 забезпечують тут спрацьовування клавіш в момент їх натискання. При запуску підпрограма спочатку очікує відпускання всіх клавіш клавіатури (блок 1). Якщо ж всі клавіші відпущені, підпрограма чекає нового натискання будь-якої з клавіш (блок 3). Відразу ж по цьому натискання йде ідентифікація натиснутою клавіші

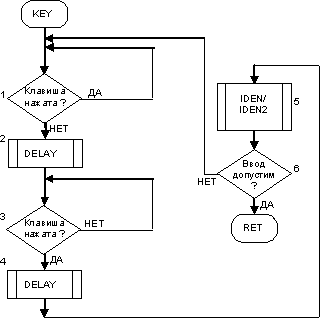


Рис.1.6.Блок-схема підпрограми IND

(Блок 5) і відбувається вихід з підпрограми з кодом KNK2 і потрібним значенням ознаки керуючої клавіші. При неприпустимому введенні спрацьовує блок 6, повертаючи управління на початок підпрограми.

Помінявши місцями блоки 1,2 і 3,4,5, можна отримати підпрограму, що має інший порядок спрацьовування клавіш - по відпускання натиснутою клавіші.

Програмні затримки DELAY (блоки 2 і 4) тривалістю 10 ... 50 мсек призначені для захисту від брязкоту. В ході відліку цих затримок мікро-ЕОМ не реагує ні на які зміни коду, що генерується клавіатурою. Реакція відновлюється лише після того, як брязкіт закінчився. Зрозуміло, що блоки 2 і 4 можна виключити з підпрограми, якщо використовується апаратна захист від брязкоту, наприклад, активні кнопки.

Розглянуту підпрограму легко модифіковані таким чином, щоб забезпечити одночасну підтримку роботи і клавіатури і семисегментного дисплея. Для цього достатньо:

1. використовувати замість блоків 2 і 4 програмної затримки DELAY підпрограму циклу індикації IND;

2. ввести виконання підпрограми IND в цикли очікування натискання і відпускання клавіш - додаткові блоки 7 і 8.

Блок - схема отриманої після модифікації підпрограми (назвемо її KEYIND) наведена на рис.1.7.

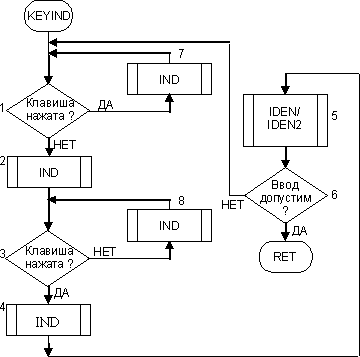


Рис.1.7.Блок-схема підпрограми KEYIND

Введення в прикладну програму підпрограми KEY (KEYIND) ілюструє блок-схема (рис.1.7).

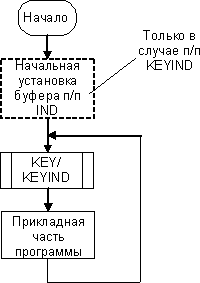


Рис.1.8.Введення підпрограм KEY (KEYIND) в прикладну програму

Введення і виконання команд управління

Користувач-оператор зазвічай відає команди Управління мікро-ЕОМ натісканням керуючих, функціональніх клавіш клавіатурі. Тому команди Управління вводяться в мікро-ЕОМ кож Як и числа - за допомога підпрограмі KEY (KEYIND). Виконання кожної Такої команди пов'язано з запуском відповідної заздалегідь підготовленої Програми Управління.

При малому чіслі керуючих клавіш допустимо організовуваті такий запуск, вікорістовуючі порівняння (по команді CPI D8) коду натіснутої клавіші з завдання и умовно передачу Управління потрібної Програмі з Z = 1.

Якщо ж керуючих клавіш багато, застосовують табличний спосіб запуску. Цей спосіб вікорістовує таблиці стартових адресу програм, Що запускаються усіма керуючий клавішамі клавіатурі. Стартові адреси програм запісані в таблиці в порядку зростання кодів KNK2 відповідніх керуючих клавіш. Код KNK2 натіснутою функціональної клавіші віявляється при цьому Покажчиком стартового адреси запускається Програми в Цій табліці. ЯКЩО вважаті, Що код KNK2 містіться в акумуляторі, функції табличного запуску програм Може Виконати фрагмент програм, уявлень на рис.1.9.

Слід підкресліті но раз, Що таблиця стартових адресу TABST повинна містіті стартові адреси Всіх наявних на клавіатурі клавіш. ЯКЩО ж натіскання будь-яких клавіш не повинно віклікаті реакції мікро-ЕОМ, слід передбачіті Спеціальну програму-заглушку, запускається цімі клавішамі.

Користувач-оператор зазвичай видає команди управління мікро-ЕОМ натисканням керуючих, функціональних клавіш клавіатури. Тому команди управління вводяться в мікро-ЕОМ також як і числа - за допомогою підпрограми KEY (KEYIND). Виконання кожної такої команди пов'язано з запуском відповідної заздалегідь підготовленої програми управління.

При малому числі керуючих клавіш допустимо організовувати такий запуск, використовуючи порівняння (по команді CPI D8) коду натиснутої клавіші з заданим і умовну передачу управління потрібної програмі з Z = 1.

Якщо ж керуючих клавіш багато, застосовують табличний спосіб запуску. Цей спосіб використовує таблицю стартових адрес програм, що запускаються усіма керуючими клавішами клавіатури. Стартові адреси програм записані в таблицю в порядку зростання кодів KNK2 відповідних керуючих клавіш. Код KNK2 натиснутою функціональної клавіші виявляється при цьому покажчиком стартового адреси запускається програми в цій таблиці. Якщо вважати, що код KNK2 міститься в акумуляторі, функції табличного запуску програм може виконати фрагмент програми, представлений на рис.1.9.

Слід підкреслити ще раз, що таблиця стартових адрес TABST повинна містити стартові адреси всіх наявних на клавіатурі клавіш. Якщо ж натискання будь-яких клавіш не повинно викликати реакції мікро-ЕОМ, слід передбачити спеціальну програму-заглушку, запускається цими клавішами.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | ; ЗАПУСК ПРОГРАММ, А = KNK2 |
|  | MOV | L,A | ; ЗАГРУЗИТЬ KNK2 В РЕГ. L |
|  | MVI | H,0 | ; ОЧИСТИТЬ РЕГ. H |
|  | LXI | D,TABST | ; ЗАГРУЗИТЬ НАЧ. АДРЕС |
|  |  |  | ; ТАБЛИЦЫ СТАРТОВЫХ |
|  |  |  | ; АДРЕСОВ В ПАРУ DE |
|  | DAD | D | ; ВЫЧИСЛИТЬ АДРЕС МЛ.БАЙТА |
|  |  |  | ; СТАРТОВОГО АДРЕСА |
|  | MOV | E,M | ; ПЕРЕСЛАТЬ МЛ. БАЙТ СТ. |
|  |  |  | ; АДРЕСА В РЕГИСТР E |
|  | INX | H | ; ВЫЧИСЛИТЬ АДРЕС СТ.БАЙТА |
|  |  |  | ; СТАРТОВОГО АДРЕСА |
|  | MOV | D,M | ; ПЕРЕСЛАТЬ МЛ. БАЙТ СТ. |
|  |  |  | ; АДРЕСА В РЕГИСТР D |
|  | XCHG |  | ; ПЕРЕСЛАТЬ СТ.АДРЕС ИЗ |
|  |  |  | ; ПАРЫ DE В ПАРУ HL |
|  | PCHL |  | ; ЗАПУСТИТЬ ПРОГРАММУ |
| TABST: | DW | ST0,ST1,…,ST7 | ; СТАРТОВЫЕ МЕТКИ |
|  |  |  | ; ЗАПУСКАЕМЫХ ПРОГРАММ |

Рис.1.9.Фрагмент програми, забезпечуючий табличний запуск управляючих програм

Вивід на семисегментний дисплей

Особливості управління семісегментним дисплеєм

Для виведення інформації мікро-ЕОМ часто використовують багаторозрядних дисплей на семисегментних світлодіодних цифрових індикаторах. Є дві схемотехнічні різновиди таких індикаторів - із загальним анодом і з загальним катодом. Принципові схеми цих індикаторів наведено на рис.1.7. У мікро-ЕОМ зазвичай використовують індикатори із загальним анодом. Цей індикатор може управлятися портом прямого виводу за схемою (рис.1.8).

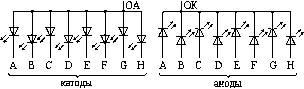


Рис.1.7.Принципові схеми семисегментних індикаторів

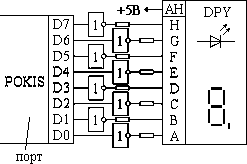


Рис.1.8.Підключення семисегментного індикатора до порту прямого вивода

Загальний анод індикатора (вивід АН) підключається до плюса джерела живлення (зазвичай 5В), а катоди світлодіодів А ... Н до виходів інверторів, часто мають відкритий колектор. Інвертори, в свою чергу, управляються портом виведення коду | з'являтися, являтися символу POKIS.

Світлодіод запалюється при потенціалі логічного нуля на виході інвертора, тобто в тому випадку, коли у відповідний розряд порту виведений одиничний біт. Резистори R обмежують на допустимому рівні струми через світлодіоди індикатора і вихідні транзистори інверторів. Лінійка інверторів може бути виключена зі схеми, якщо порт POKIS має достатньо потужні інверсні виходи. Може використовуватися і порт з прямими виходами, але тоді в нього слід виводити не код | з'являтися, являтися символу KIS, а його інверсію.

Схему багато розрядного дисплея можна отримати, просто повторивши потрібне число раз схему управління одиничним індикатором. Така схема (її називають схемою зі статичним керуванням) повинна містити стільки портів виводу, яка розрядність дисплея. Тому на практиці її використовують рідко. Як правило, використовується схема динамічного або мультиплексного управління багато розрядним дисплеєм. Типовий варіант такої схеми для шестирозрядного дисплею представлений на рис.1.9.

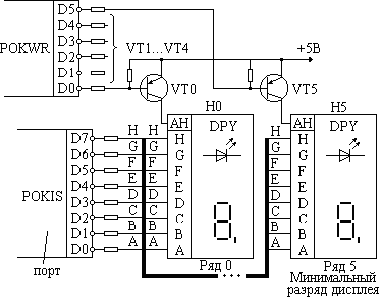


Рис.1.9.Шестирозрядний семисегментний дисплей з динамічним керуванням

Схему створюють шість ідентичних вертикальных рядків, кожний з котрих має по семисегментному індикатору Н0…Н5 і транзисторному ключу VT0…VT5. Рядкі пронумеровані зліва направо цифрами від 0 до 5. Схемою керують 2 порти прямого виводу POKWR і POKIS.

В порт POKWR виводиться код вибору ряду KWR. Одиничний біт, виведений в цей порт, включає, активує відповідний ряд схеми. В порт POKIS виводиться код індицируємого символа KIS. Цей символ буде засвічений на індикаторі активного ряду.

Типовий режим роботи дисплею пояснює слідуючі часові діаграми (рис.1.10).

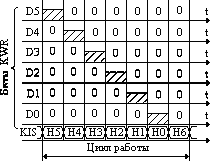


Рис.1.10. Часові діаграми роботи семисегментного дисплею з динамічним управлінням

З діаграм видно, що повний цикл роботи дисплею складається з шести послідовних інтервалів. На кожному з них кодом KWR вибираються, активується, включається тільки один індикатор, а інші погашені. Ім’я активного індикатора на кожному з інтервалів вказано на діаграмі кода KIS. Неодночасність роботи індикаторів дисплею є замітною користувачу тільки при достатньо великій тривалості циклу. Якщо ж цей час меньше 1/30…1/50 сек, у користувача створюється повна ілюзія їх одночасної роботи.

Програмна підтримка роботи семисегментного дисплею з дінамічним керуванням

Один повний цикл керування семисегментним дисплеєм формують за допомогою спеціальної підпрограми. Блок-схема і текст цієї підпрограми представлений відповідно на рис.1.11 и рис.1.12. Для незперервної індикації рядку символів, занесених до ячейки S0…S5 достатньо використати підпрограму IND в тілі незкінченого циклу. Відповідний фрагмент програми представлений на рис.1.12.

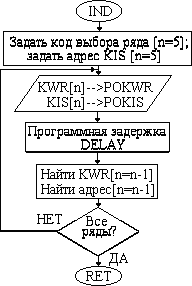


Рис.1.11.Блок-схема підпрограми IND n – номер рядку на схемі дисплею з динамічним управлінням

«Рядок , який біжить»

Іноді число знакомест дисплея виявляється менше числа символів, в виведеної рядку. У цих випадках інформація може виводитися на дисплей в режимі «біжучого рядка». Блок-схема програми, що забезпечує цей режим виводу наведена на ріс.1.12.

Програма використовує буфер для семисегментних кодів, що виводяться на дисплей, що має структуру, представлену на ріс.4.28. Символи рядка, що виводиться записуються в N послідовних комірок буфера, починаючи з клітинки S0, після чого управління передається на мітку START0. Перші NN циклів індикації на дисплеї відображаються шість лівих символів рядка. Далі виконується циклічна пересилка (поворот інформації в буфері). При повороті інформація пересилається таким чином, щоб код з комірки S1 опинився в осередку S0, з S2 - в S1 і так далі. Код з осередку S0 пересилається осередок буфера з самим старшим адресою SH. Цю пересилання зручніше виконати в два етапи - спочатку переслати код з S0 в робочу комірку S00 і лише потім, коли буде пересланий код з останньої клітинки буфера SH, занести в цей осередок код з комірки S00. Поворот буфера ілюструє діаграма (ріс.1.14). На ній жирним шрифтом виділено

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | ; ПОДПРОГРАММА IND |
| POKWR | EQU | … | ; АДРЕС POKWP |
| POKIS | EQU | … | ; АДРЕС POKIS |
| KWRS | EQU | 00100000B | ; KWR [5] |
| IND: | MVI | B, KWR5 | ; ЗАДАТЬ KWR [n=5] |
|  | LXI | H, S5 | ; ЗАДАТЬ АДРЕС KIS [n=5] |
| M0: | MOV | A, B | ; ВЫВЕСТИ KWR [N] В |
|  | OUT | POKWR | ; ПОРТ POKWR |
| MOV | A, M |  | ; ВЫВЕСТИ KIS [n] В |
| OUT | POKIS |  | ; ПОРТ POKIS |
|  | CALL | DELAY | ; ПРОГРАММНАЯ ЗАДЕРЖКА |
|  | MOV | A, B | ; НАЙТИ KWR [n = n - 1] |
|  | RRC |  | ; |
|  | MOV | B, A | ; |
|  | DCX | H | ; НАЙТИ АДРЕС KIS [n = n - 1] |
|  | JNC | M0 | ; ВСЕ РЯДЫ ? НЕТ, ПОВТОРИТЬ |
|  | RET |  | ; ИНАЧЕ ВЫЙТИ В ОСНОВНУЮ |
|  |  |  | ; ПРОГРАММУ |
| DELAY: | … |  | ; ПРОГРАММНАЯ ЗАДЕРЖКА |
| S0: | DS | 1 | ; БУФЕР ИЗ ШЕСТИ ЯЧЕЕК |
|  |  |  | ; СЕМИСЕГМЕНТНОЙ |
|  |  |  | ; ИНДИКАЦИИ |
| S1: | DS | 1 | ; |
| S2: | DS | 1 | ; |
| S3: | DS | 1 | ; |
| S4: | DS | 1 | ; |
| S5: | DS | 1 | ; |

Рис.1.12.Текст підпрограми IND

|  |  |  |
| --- | --- | --- |
|  | … |  |
| MM: | CALL | IND | |
|  | JMP | MM | |
|  | … |  |

Рис.1.13.Фрагмент програми, яка забезпечує безперервну індикацію рядка символів

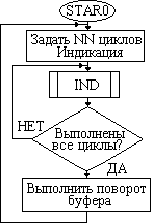


Рис.1.14.Блок-схема програми, яка забезпечує вивід в режимі «бегущей рядка»

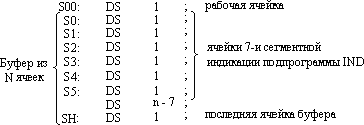


Рис.1.15.Структура буфера програми, яка забезпечує вивід в режимі бегущей рядка

символи, семисегментні коди яких знаходяться в буфері. Числа близько стрілок задають порядок пересилання кодів в процесі повороту. Після повороту буфера знову виконується NN циклів індикації. Легко бачити, що при цьому символи на дисплеї будуть послідовно зміщуватися вліво і рядок на ньому «побіжить». Швидкість її руху визначається числом циклів NN циклів індикації. Чим більше NN, тим повільніше біжить рядок.

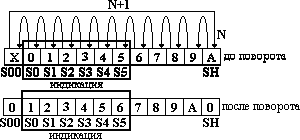


Рис.1.16.Поворот буфера

Для поворотe буферe маючого N комірок можна скористатися фрагментом програми, яка представлена на рис.1.17.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | ; ПОВОРОТ БУФЕРА |
| N | EQU | … | ; ЧИСЛО СИМВОЛОВ В СТРОКЕ |
|  | LXI | H, S0 | ; ЗАНЕСТИ В ПАРУ HL НАЧ. |
|  |  |  | ; АДРЕС ИСТОЧНИКА КОДА |
|  | LXI | D, S00 | ; ЗАНЕСТИ В ПАРУ DE НАЧ. |
|  |  |  | ; АДРЕС ПРИЕМНИКА КОДА |
|  | MVI | B, N | ; ВЫПОЛНИТЬ N ЦИКЛОВ |
| M0: | MOV | A, M | ; ПЕРЕСЛАТЬ КОД ИЗ |
|  | STAX | D | ; ИСТОЧНИКА В ПРИЕМНИК |
|  | INX | H | ; МОДИФИЦИРОВАТЬ АДРЕСА |
|  | INX | D | ; ИСТОЧНИКА И ПРИЕМНИКА |
|  | DCR | B | ; ВСЕ ЦИКЛЫ? |
|  | JNZP | M0 | ; НЕТ, ПОВТОРИТЬ, ИНАЧЕ |
|  | LDA | S00 | ; ПЕРЕСЛАТЬ КОД |
|  | STA | SH | ; ИЗ S00 В SH |
|  | … |  |  |

Рис.1.17.Фрагмент програми, яка забезпечує поворот буферу

Перетворення двійкових кодів в семисегментні

Висновок на семисегментний дисплей результатів розрахунку або кодів, набраних на клавіатурі, зазвичай пов'язаний з переведенням двійкових кодів шістнадцятиричних цифр в семисегментні. Такий переклад виконується за допомогою підпрограми табличного перетворення. Існує кілька видів такого перетворення. Тут доцільно застосувати простий вигляд, який використовує вихідний двійковий код в якості покажчика адреси свого семисегментного еквівалента в таблиці семисегментних кодів.

Перетворення ілюструє підпрограма BIS7 (рис.1.18), яка перетворює в семисегментний код молодшу тетраду байта, що знаходиться в акумуляторі.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | ; ПОДПРОГРАММА BIS7 |
| MASK | EQU | 0FH | ; МАСКА МЛАДШЕЙ ТЕТРАДЫ |
| BIS7: | ANI | MASK | ; ВЫДЕЛИТЬ МЛ. ТЕТРАДУ |
|  |  |  | ; ВХОДНОГО КОДА |
|  | MVI | H, 0 | ; ОЧИСТИТЬ РЕГИСТР H |
|  | MOV | L, A | ; ЗАГРУЗИТЬ МЛ. ТЕТРАДУ ВХ |
|  |  |  | ; КОДА В РЕГИСТР L |
|  | LXI | D, TAB7 | ; ЗАГРУЗИТЬ НАЧ. АДР.ТАБЛИЦЫ |
|  |  |  | ; СЕМИСЕГМЕНТНЫХ КОДОВ |
|  |  |  | ; В ПАРУ DE |
|  | DAD | D | ; ВЫЧИСЛИТЬ АДРЕС ВЫХ. КОДА |
|  | MOV | A, M | ; ЗАГРУЗИТЬ ВЫХ. КОД В РЕГ. А |
|  | RET |  | ; ВОЗВРАТИТЬСЯ В ОСНОВНУЮ |
|  |  |  | ; ПРОГРАММУ |
| TAB7: | DB | …, …, …, …, …, …, …, … | ; ТАБЛИЦА СЕМИСЕГМЕНТНЫХ |
|  | DB | …, …, …, …, …, …, …, … | ; КОДОВ ЦИФР 0…F |

Рис.1.18. Текст підпрограми BIS7

Початковий (базовий) адреса таблиці TAB7 складається в підпрограмі з вихідним кодом 0 ... F. При цьому в регістровий парі HL виходить адресу відповідного семисегментного коду. Цей код витягується з таблиці TAB7 командою MOV A, M і поміщається в акумулятор.

**Завдання.**

1.Створіть функціональну схему мікропроцесорної системи з клавіатурою та індикацією та схему в графічному редакторі.

2.Напишіть програму опитування клавіатури та виводу даних на індикатор.

3.Вивід інформації на семигментний індикатор.

4.Вивід інформації на матричний індикатор.

**Контрольні питання**

1.Що таке коди клавіатури?

2.Що таке ASC ІІ коди?

Схема клавіатури та індикації комплексу



Таблиця кодів знаків які виводяться на індикацію(семисегментний індикатор)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Знаки | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Код |
| A | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 77h |
| b | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 7ch |
| C | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 39h |
| d | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 5eh |
| E | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 79h |
| F | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 71h |
| G | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 3dh |
| H | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 76h |
| I | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 06h |
| J | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0eh |
| L | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 38h |
| П | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 37h |
| O | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3fh |
| P | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 73h |
| S | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 6dh |
| U | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 3eh |
| Y | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 6eh |
| z | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5bh |
| - | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 40h |
| . | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 80h |