**РАБОТА 2. ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА**

**ИНФОРМАЦИИ В МИКРОКОНТРОЛЛЕРЕ**

***Цель работы:***изучить организацию ввода и вывода информации через параллельные порты микроконтроллера, особенности работы с отдельными линиями параллельных портов, специфику использования отдельных команд микроконтроллера и его режимов адресации.

**Введение**

В силу специфики области своего применения, однокристальные микроконтроллеры обладают широкими возможностями по вводу и выводу информации. Микроконтроллеры семейства МК‑51 для этих целей имеют один последовательный порт и четыре параллельных 8-разрядных порта P0…P3., каждая линия которых может индивидуально настраиваться на ввод либо вывод информации.

Каждый из параллельных портов, помимо возможности его использования в качестве универсального порта ввода-вывода, несет дополнительную функциональную нагрузку в зависимости от кон​фи​гурации микропроцессорной системы и от конкретного типа БИС.

Через порт P0 при работе с внешней памятью программ (ВПП) и внешней памятью данных (ВПД) выводится младший байт адреса (А7…А0), вводится байт команды из ВПП, вводится и выводится байт данных при работе с ВПД (все эти операции мультиплексированы во времени и поддерживаются аппаратно), задаются данные при программировании внутреннего ППЗУ в тех БИС, где оно предусмотрено, и читается содержимое внутренней памяти программ.

Через порт P1 вводится младший байт адреса (А7…А0) при программировании внутреннего ППЗУ и чтении внутренней памяти программ.

Порт P2 используется для вывода старшего байта адреса (А15…А8) при обращении к ВПП и ВПД (также мультиплексно во времени) и ввода разрядов А12…А8 адреса при программировании внутреннего ППЗУ и чтении внутренней памяти программ.

Каждый из разрядов порта Р3 используется для выполнения какой-либо альтернативной функции (см. рис.4.). Для выполнения этим портом альтернативных функций необходимо, чтобы защелки соответствующих разрядов содержали "1".

Каждый вывод портов Р1, Р2, Р3 может независимо от других настраиваться как вход или как выход. Для использования вывода в качестве входа необходимо, чтобы его защелка содержала "1". Эти порты в режиме "оборваный вход" имеют уровень "1", благодаря чему их иногда называют квазидвунаправленными.

При использовании порта Р0 в качестве выходного порта общего назначения необходимо устанавливать на его выводах внешние резисторы для задания уровня "1". Порт Р0 является в чистом виде двунаправленным.

Напомним, что по сигналу аппаратного сброса защелки всех портов устанавливаются в "1" (см. табл. 2).

**Постановка задачи и варианты ее решения**

В зависимости от варианта, микроконтроллер осуществляет ввод (ВВ) либо вывод (ВЫВ) информации.

Данные представляют собой десятичные цифры, записанные либо в упакованном (УП), либо в распакованном (РАСП) формате. В первом случае в каждом байте хранятся по две десятичные циф​​​​ры, старшая занимает левую тетраду, а младшая - правую. Во втором случае каждая цифра занимает лишь младшую тетраду бай​та, старшая тетрада при обмене не должна меняться.

Начальный адрес массива, из которого выводятся или в ко​то​рый вводятся данные, либо хранится в регистре R0 текущего банка дан​ных, либо имеет фиксированное значение, указанное в таблице вариантов. Длина массива, задаваемая в цифрах, либо хранится в регистре R1, либо равна первой цифре массива (ПЦ), либо конец обмена определяется сигналом низкого уровня на входе P1.7. В первом случае цифра, определяющая длину массива, входит в состав массива и должна передаваться вместе с массивом. Обмен осуществляется параллельным (ПР) по каналам P1.3...P1.0 либо последовательным по каналу P1.0 кодом. Последовательная пе​ре​дача может осу​щес​т​вляться старшими (ПССТ) либо младшими (ПСМЛ) битами вперед.

Инициатором обмена выступает микроконтроллер (МК) либо внешнее устройство (ВУ). В первом случае каждая операция обмена начинается с выдачи микроконтроллером сигнала запроса к ВУ. Обмен осуществляется после приема от ВУ сигнала подтверждения, при поступлении которого МК производит обмен, снимает запрос, ждет снятия сигнала подтверждения со стороны ВУ и затем продолжает работу.

Во втором случае прежде, чем начать обмен МК ожидает сигнала запроса от ВУ, затем выставляет сигнал готовности к обмену, производит обмен, снимает сигнал готовности, ждет снятия запроса от ВУ, после чего продолжает работу.

Обмен управляющими сигналами между МК и ВУ осуществляется при каждой передаче. ВУ передает сигналы в МК (подтверждение или запрос) по линии Р1.4. МК передает сигналы к ВУ (запрос или готовность) по линии Р1.5. Активное значение сигнала указывается в варианте задания (Н - высокий, L - низкий).

После завершения передачи массива управление передается на начало программы.

Отметим некоторые моменты, связанные с разработкой программ данной лабораторной работы.

Так как программа работает с некоторым массивом данных, то целесообразно использовать косвенную адресацию элементов этого массива (более сложные типы адресации данных, которые можно было бы применить в этом случае, в системе команд микроконтроллера отсутствуют). При этом начальный адрес массива может храниться лишь в регистре R0 или R1 и должен увеличиваться каждый раз после обработки очередного байта памяти (для распакованных чисел – это обработка очередной цифры, а для упакован​ных – обработка двух цифр).

Неоднократно используемые в программе действия удобно оформить в виде подпрограмм, что делает программу более наглядной и короткой.

Для приема цифры, поступающей в последовательном коде по одной из линий параллельного порта, можно использовать команды циклического сдвига аккумулятора с учетом бита переноса С (битового аккумулятора). Схема выполнения этих команд представлена на рис.2.1.

В отдельных вариантах задания оказывается удобным применение специальной команды SWAP A, которая меняет местами тетрады аккумулятора A(3…0) ↔ A(7…4).

**Порядок подготовки к лабораторной работе**

1.​ Изучить пример подготовки к выполнению работы.

2.​ Составить программу на языке ассемблера МК-51.

3.​ Подготовить тестовые примеры для отладки программы.

**Порядок выполнения лабораторной работы**

1.​ Выполнить отладку программы.

2.​ Продемонстрировать работу отлаженной программы пре​подавателю.

3.​ Ответить на вопросы преподавателя.

**Пример подготовки**

**к выполнению лабораторной работы**

Подготовку к лабораторной работе разберем на примере двух вариантов заданий, существенно отличающихся организацией обмена: параллельным и последовательным кодами.

***Вариант 31.*** Согласно варианту мик​​роконтроллер осуществляет ввод pаспакованных чисел последовательным кодом по линии Р1.0 стаpшими pазpядами впеpед. Начальный адpес массива перед выполнением этой задачи уже хранится в pегистpе R0. Длина массива - пеpвая пpинятая цифpа. После ввода каждого бита МК выдает сигнал низкого уpовня по линии Р1.4. Разpешение ввода - сигнал низкого уpовня по линии Р1.5. Инициатор обмена –внешнее устройство.

Программа реализации данного варианта задания с необходимыми комментариями приведена ниже.

*Программа*

.ORG 0H

AJMP BEGIN

.ORG 30H

;Р1.4 и Р1.0-на ввод,Р1.5 -неактивный (высокий) уpовень

BEGIN: MOV P1,#00110001b

ACALL SEND ; вызов п/п ввода цифры в аккумулятор

;Ввод пеpвой цифpы (длины массива) в R3

MOV R3,A ;длина массива - в R3

ACALL WRITE ;вызов п/п записи цифры в память

M2: DJNZ R3,M1 ;проверка на конец ввода массива

AJMP BEGIN

M1: ACALL SEND ;вызов п/п ввода цифры

ACALL WRITE ;вызов п/п записи цифры в память

AJMP M2

;подпрограмма ввода цифры в А3…А0

SEND: MOV R2,#4 ;счетчик бит в цифpе

CLR A ;A=0

IN: JNB P1.5,IN ;ожидание запроса от ВУ

CLR P1.4 ;выдача сигнала готовности МК

MOV C,P1.0 ;ввод бита

SETB P1.4 ;снятие сигнала готовности МК

JNB P1.5,$ ;ожидание снятия запроса от ВУ

RLC A ;A<0>=(введенный бит)

DJNZ R2,IN ;конец цикла ввода бита

RET

; подпрограмма записи введенной цифры в массив

WRITE: MOV R4,A ;сохранение введенной цифры

MOV A,@R0 ;А=(старое значение байта)

ANL A,#11110000b ;(A3...A0)=0

ORL A,R4 ;формирование распакованной цифры

MOV @R0,A ;запись цифры в массив

INC R0 ;i=i+1

RET ;возврат из подпрограммы

.END

***Вариант 32.*** Согласно варианту мик​​​роконтроллер осуществляет вывод упакованных чисел параллельным кодом по каналам P1.3...Р1.0. Начальный адpес массива -10H. Длина массива – его пеpвая цифpа. Инициатор обмена – МК. Запрос от МК - сигнал низкого уpовня по каналу Р1.4. Подтверж​дение от ВУ - сигнал высо​кого уpовня по каналу Р1.5.

Программа реализации данного варианта задания с необходимыми комментариями приведена ниже.

*Программа*

.ORG 0H

AJMP START ;переход на начало программы

.ORG 30H

START: MOV P1,#00110000b ;Р1.5 - на ввод, Р1.4 -высокий

MOV R0,10H ;R0 - байт, содержащий длину массива

ANL R0,#0FH ;выделение длины массива

MOV R1,#10H ;R1 - начальный адрес массива (i=0)

OUTB: MOV A,@R1 ;А=аi

ACALL SEND ;переход на п/п вывода цифры

DJNZ R0,OUTDG ;если R00, то на вывод первой цифры

AJMP START ;возврат на начало программы

OUTDG: MOV A,@R1 ;А=(байт)i

SWAP A ;обмен тетрадами аккумулятора

ACALL SEND ;переход на п/п вывода цифры

INC R1 ;i=i+1

DJNZ R0,OUTB ;если R00, то на вывод следующего байта

AJMP START ;возврат на начало программы

; подпрограмма вывода цифры

SEND: ANL A,#00001111b ;очистка A.7...A.4

ANL P1,#11110000b; очистка P1.3...P1.0

CLR P1.4 ;запрос от МК

JNB P1.5,$ ;ожидание подтверждения от ВУ

ORL P1,A ;вывод цифры

SETB P1.4 ;снятие запроса

JB P1.5,$ ;ожидание снятия подтверждения

RET ;возврат из п/п

.END

**Варианты заданий**

| **№**  **варианта** | **Вид**  **числа** | **ВВ/**  **ВЫВ** | **Вид**  **обмена** | **Начальный**  **адрес** | **Длина**  **массива** | **Инициатор** | **Уровни**  **сигналов** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **МК** | **ВУ** |
|  |  |  |  |  |  |  |  |  |
| 12 | УП | ВЫВ | ПССТ | 20H | P1.7 | ВУ | L | L |