



ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

Электронный учебно-методический комплекс

Методические указания по выполнению лабораторных работ

Програмное обеспечение для выполнения лабораторных работ

Лабораторная работа №1

Лабораторная работа №2

Лабораторная работа №3

Лабораторная работа №4

Лабораторная работа №5

Лабораторная работа №6

Лабораторная работа №7

Лабораторный практикум

ПечататьЗакры

Лабораторная работа №4

1 Цель и содержание 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.

2 Постановка задачи

В силу специфики области своего применения, однокристалльные 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 конфигурации микропроцессорной системы.

Каждый вывод портов *P1*, *P2*, *P3* может независимо от других настраиваться как вход или как выход. Для использования вывода в качестве входа необходимо, чтобы его защелка содержала «1».

2.1 Алгоритм работы

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

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

Начальный адрес массива, из которого выводятся или в который вводятся данные, либо хранится в регистре *R0* текущего банка данных, 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*. 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 передача может осуществляться 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. *МК* передает сигналы в *МК* (подтверждение или запрос) по линии *P1.4*. *МК* передает сигналы к *ВУ* (запрос или готовность) по линии *P1.5*. Активное значение сигнала указывается в варианте задания (*H* — 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 упакованных — 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 (битового аккумулятора). Схема выполнения этих команд представлена на рисунке 1.

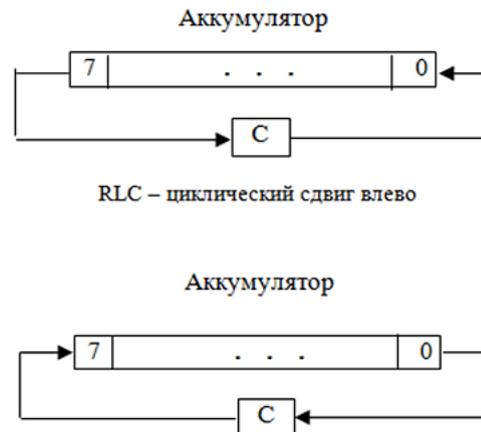


Рисунок 1 — Схема выполнения операции циклического сдвига с учетом бита переноса

В отдельных вариантах задания оказывается удобным применение специальной команды *SWA*, которая меняет местами тетрады аккумулятора $A(3...0) \leftrightarrow A(7...4)$.

2.2 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.

2.3 Порядок выполнения лабораторной работы

1. Выполнить отладку программы.
2. Продемонстрировать работу отлаженной программы преподавателю.
3. Ответить на вопросы преподавателя.

2.4 Пример выполнения работы

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

Вариант 31. Согласно варианту микроконтроллер осуществляет ввод распакованных чисел последовательным кодом по линии *P1.0* старшими разрядами вперед. Начальный адрес массива перед выполнением этой задачи уже хранится в регистре *R0*. Длина массива — первая принятая цифра. После ввода каждого бита *МК* выдает сигнал низкого уровня по линии *P1.4*. Разрешение ввода — сигнал низкого уровня по линии *P1.5*. Инициатор обмена — внешнее устройство.

Программа реализации данного варианта задания с необходимыми комментариями приведена ниже.

Программа

```
.ORG 0H
    AJMP    BEGIN
.ORG 30H
;P1.4 и P1.0-на ввод,P1.5 – неактивный (высокий) уровень

BEGIN: MOV    P1,#00110001b
        ACALL SEND                ; вызов п/п ввода цифры в аккумулятор
        ;Ввод первой цифры (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 A3..A0
SEND:  MOV    R2,#4                ;счетчик бит в цифре
        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      ;A=(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 микроконтроллер осуществляет вывод упакованных чис параллельным кодом по каналам P1.3...P1.0. Начальный адрес массива — 10H. Длина массива — ε первая цифра. Инициатор обмена — МК. Запрос от МК — сигнал низкого уровня по каналу P1 Подтверждение от ВУ — сигнал высокого уровня по каналу P1.5.

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 ;P1.5 – на ввод, P1.4 – 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 ;A=ai
ACALL SEND ;переход на п/п вывода цифры
DJNZ R0,OUTDG ;если R0≠0, то на вывод первой цифры
AJMP START ;возврат на начало программы
OUTDG: MOV A,@R1 ;A=(байт)i
SWAP A ;обмен тетрадами аккумулятора
ACALL SEND ;переход на п/п вывода цифры
INC R1 ;i=i+1
DJNZ R0,OUTB ;если R0≠0, 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

```

3 Варианты заданий

№ варианта	Вид числа	ВВ/ ВВВ	Вид обмена	Начальный адрес	Длина массива	Инициатор	Уровни сигналов	
							МК	ВУ
1	РАСП	ВВВ	ПР	R0	R1	МК	Н	Н
2	УП	ВВ	ПССТ	10H	R1	ВУ	Н	L
3	РАСП	ВВ	ПР	20H	ПЦ	МК	L	Н
4	УП	ВВВ	ПСМЛ	R0	ПЦ	ВУ	L	L
5	РАСП	ВВВ	ПССТ	10H	P1.7	МК	Н	Н
6	УП	ВВ	ПСМЛ	20H	P1.7	ВУ	Н	L
7	РАСП	ВВ	ПССТ	R0	R1	МК	L	Н
8	УП	ВВВ	ПР	10H	R1	ВУ	L	L
9	РАСП	ВВВ	ПСМЛ	20H	ПЦ	МК	Н	Н
10	УП	ВВ	ПР	R0	ПЦ	ВУ	Н	L
11	РАСП	ВВ	ПСМЛ	10H	P1.7	МК	L	Н
12	УП	ВВВ	ПССТ	20H	P1.7	ВУ	L	L
13	РАСП	ВВВ	ПР	R0	R1	МК	Н	Н
14	УП	ВВ	ПССТ	10H	R1	ВУ	Н	L
15	РАСП	ВВ	ПР	20H	ПЦ	МК	L	Н
16	УП	ВВВ	ПСМЛ	R0	ПЦ	ВУ	L	L
17	РАСП	ВВВ	ПСМЛ	10H	R1	МК	Н	Н
18	УП	ВВ	ПССТ	20H	R1	ВУ	Н	L
19	РАСП	ВВ	ПССТ	R0	ПЦ	МК	L	Н
20	УП	ВВВ	ПР	10H	ПЦ	ВУ	L	L

№ варианта	Вид числа	ВВ/ВЫВ	Вид обмена	Начальный адрес	Длина массива	Инициатор	Уровни сигнала	
							МК	ВУ
21	РАСП	ВЫВ	ПСМЛ	20H	P1.7	МК	H	H
22	УП	ВВ	ПР	R0	P1.7	ВУ	H	L
23	РАСП	ВВ	ПСМЛ	10H	R1	МК	L	H
24	УП	ВЫВ	ПССТ	20H	R1	ВУ	L	L
25	РАСП	ВЫВ	ПР	R0	ПЦ	МК	H	H
26	УП	ВВ	ПССТ	10H	ПЦ	ВУ	H	L
27	РАСП	ВВ	ПР	20H	P1.7	МК	L	H
28	УП	ВЫВ	ПСМЛ	R0	P1.7	ВУ	L	L
29	РАСП	ВЫВ	ПССТ	10H	R1	МК	H	H
30	УП	ВВ	ПСМЛ	20H	R1	ВУ	H	L
31	РАСП	ВВ	ПССТ	R0	ПЦ	МК	L	L
32	УП	ВЫВ	ПР	10H	ПЦ	ВУ	L	H

4 Содержание отчета

Отчет о лабораторной работе должен содержать:

- титульный лист;
- цель и задачи работы;
- скриншоты экрана среды разработки;
- текст программы на ассемблере с комментариями;
- при необходимости структурные схемы и временные диаграммы;
- выводы по работе.

[← предыдущая](#)

[следующая →](#)