

Саратовский государственный университет им. Н.Г.Чернышевского

А.А.Семёнов

ИЗУЧЕНИЕ БИС  
ПАРАЛЛЕЛЬНОГО  
ПЕРИФЕРИЙНОГО  
АДАПТЕРА

Учебное пособие  
для студентов факультета компьютерных наук  
и информационных технологий  
и факультета nano- и биомедицинских технологий

Издательство Саратовского университета  
2006

УДК 621.382-181(075.8)

ББК 32.844.1 я73

С30

**Семёнов А.А.**

С30 Изучение БИС параллельного периферийного адаптера: Учеб. пособие для студ. фак. компьютерных наук и информационных технологий и фак. nano- и биомедицинских технологий – Саратов: Изд-во Саратов. ун-та, 2006. – 44 с.: ил.

ISBN        -        -        -       

Учебное пособие представляет собой руководство к практическим занятиям по курсам "Устройство и применение микропроцессоров" и "Микропроцессорные системы". Содержит теоретическое описание материала, знание которого необходимо при выполнении лабораторных и практических работы по изучению БИС параллельного периферийного адаптера K580VB55.

Для студентов, обучающихся по направлениям "Электроника и микроэлектроника", "Физика полупроводников и диэлектриков" специальностям "Вычислительные машины, комплексы, системы и сети", "Микроэлектроника и полупроводниковые приборы".

Рекомендуют к печати:

Кафедра физики твердого тела факультета  
нано- и биомедицинских технологий  
Саратовского государственного университета  
Доктор физико-математических наук В.Б. Байбурин,  
доктор физико-математических наук А.В. Скрипаль

УДК 621.382-181 (075.8)

ББК 32.844.1 я73

ISBN        -        -        -       

© Семёнов А.А., 2006

## ОГЛАВЛЕНИЕ

|  |    |
|--|----|
| <b>В в е д е н и е</b> .....   | 4  |
| <b>1. Теоретическая часть</b> .....  | 5  |
| 1.1. Архитектура параллельного периферийного адаптера (ППА) КР580ВВ55....          | 5  |
| 1.2. Подключение ППА к магистралям микропроцессорной системы.....                  | 6  |
| 1.3. Начальная инициализация адаптера.....   | 9  |
| 1.4. Режимы работы адаптера.....   | 10 |
| 1.5. Основы программирования микропроцессорных систем на языке<br>ассемблера ..... | 16 |
| 1.5.1. Формулировка задачи.....  | 17 |
| 1.5.2. Решение задачи.....   | 18 |
| 1.5.3. Проектирование программы.....   | 19 |
| 1.5.4. Проектирование модулей.....   | 19 |
| 1.5.5. Логическая структура ассемблерной программы.....                            | 20 |
| <b>2. Экспериментальная часть</b> .....  | 26 |
| 2.1. Программно-аппаратные средства.....   | 26 |
| 2.2. Практические задания.....   | 28 |
| 2.3. Порядок выполнения заданий.....   | 39 |
| <b>Список литературы</b> .....   | 39 |
| <b>Приложение. Система команд микропроцессора К580ВМ80</b> .....                   | 40 |

## ВВЕДЕНИЕ

В практике управления микропроцессорной системой (МПС) периферийными устройствами встречаются ситуации, когда управляющие сигналы передаются по нескольким линиям одновременно. Простейшим примером является управление 8–сегментным 8–разрядным индикатором, когда 16 сигнальных линий одновременно задают восьмисегментный код очередного отображаемого разряда дисплея и его позицию в индикаторной матрице. Совокупность программных и аппаратных средств управления индикатором в данном случае образует параллельный интерфейс, определяющий параллельный способ обмена данными МПС с внешним устройством (ВУ).

В качестве аппаратных средств интерфейса в самом простом случае могут применяться регистры с параллельной записью информации, включенные в состав МПС как устройства ввода/вывода (УВВ). Сигнал записи микропроцессора (МП), подключенный к регистрам в качестве стробирующего, позволяет зафиксировать в регистрах состояние бит отдельных разрядов магистрали данных (МД). Выходные линии регистров осуществляют передачу данных внешнему устройству в параллельном коде.

В более сложных ситуациях асинхронное управление внешним устройством наряду с сигналами данных может потребовать дополнительных сигналов (сигналов квитирования), определяющих направление передачи, подтверждение истинности данных, состояние ВУ и т.д. Сложный процесс асинхронного обмена данными между МПС и ВУ, а также требования наиболее рационального распределения времени работы системы приводят к необходимости разработки специальных интерфейсных устройств при организации обмена данными. Задача таких устройств состоит в приеме данных по сигналам управления, формировании сигналов, указывающих на наличие данных в устройстве, выдаче данных по сигналам управления, а также формировании сигналов, указывающих на готовность устройства к приему новых данных.

Технология производства больших интегральных схем (БИС) позволяет создавать универсальные многофункциональные интерфейсные устройства ввода/вывода параллельной информации. Такие устройства могут быть программируемыми и применяться для двунаправленной передачи данных и сигналов управления. МПС информируется о готовности устройства к обмену данными, как правило, по сигналам прерывания. Обмен данными осуществляется путем обращения микропроцессора (МП) к устройству ввода/вывода как к ячейке памяти или ВУ. Таким образом, интерфейсные устройства ввода/вывода параллельной информации позволяют

согласовать во времени процесс обмена данными между МПС и ВУ при рациональном использовании времени работы микропроцессора.

В составе микропроцессорного комплекта К580 функции программируемого устройства ввода/вывода параллельной информации различного формата выполняет БИС *параллельного периферийного адаптера К580ВВ55А (КР580ВВ55, К580ИК55)*. Микросхема предназначена для организации ввода/вывода информации в параллельном коде и позволяет реализовать большинство известных протоколов обмена по параллельным каналам. БИС ППА обычно используется для сопряжения микропроцессорной системы со стандартным периферийным оборудованием (цифровым дисплеем, принтером, телетайпом, различными внешними накопителями) через три 8-разрядных канала, предназначенных для передачи как данных, так и управляющих сигналов. Микросхема может быть использована для организации синхронного (при одинаковом быстродействии МП и ВУ) и асинхронного (с более медленными ВУ) обмена, а также обмена в режиме прерывания.

## 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### 1.1. АРХИТЕКТУРА ПАРАЛЛЕЛЬНОГО ПЕРИФЕРИЙНОГО АДАПТЕРА К580ВВ55

Структурная схема параллельного периферийного адаптера представлена на рисунке 1.

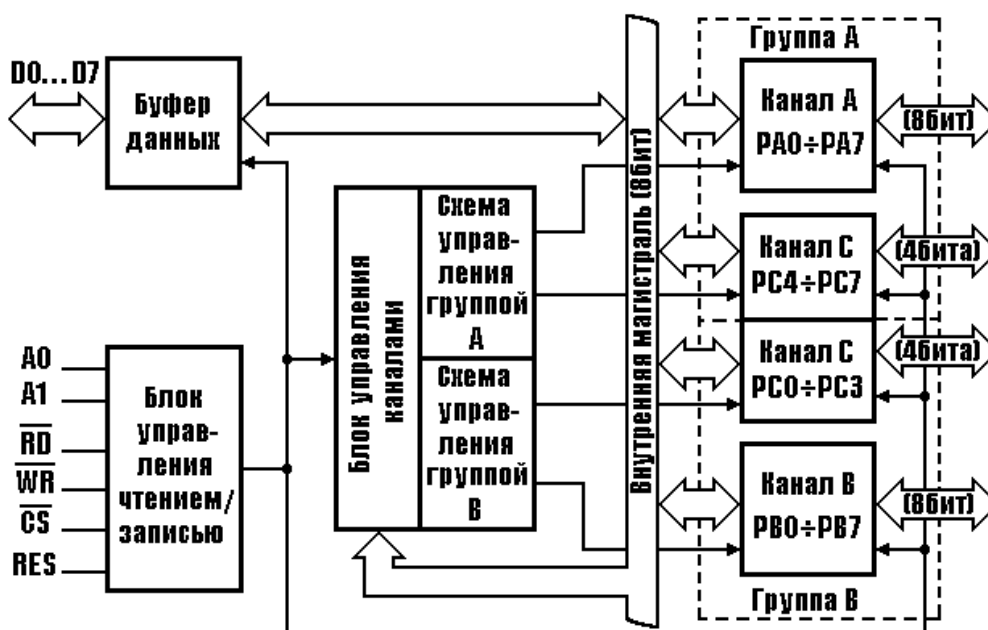
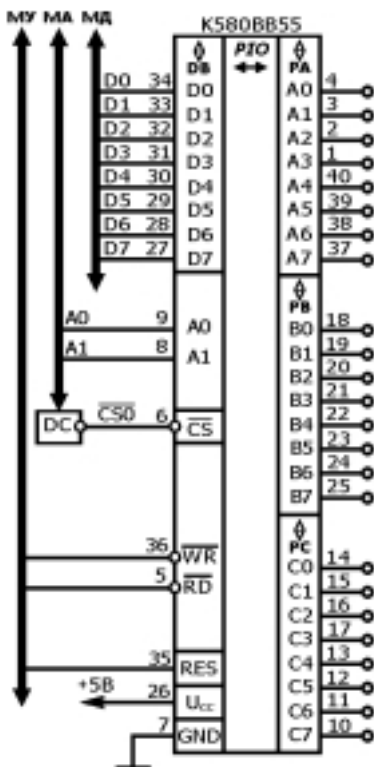


Рис. 1. Структура параллельного периферийного адаптера К580ВВ55

В состав БИС входят: двунаправленный 8-разрядный *буфер данных*, связывающий *внутреннюю 8-разрядную магистраль* ППА с системной магистралью данных; *блок управления чтением/записью*, обеспечивающий управление внешними и внутренними передачами данных, управляющих слов и информации о состоянии ППА; три 8-разрядных *канала ввода/вывода* (PORT A, PORT B, PORT C) для обмена информацией с внешними устройствами, объединяемые в две *группы* – *A* и *B*; *блок управления каналами*, состоящий из *схемы управления группой A*, вырабатывающей сигналы управления *каналом A* и старшими разрядами *канала C* (PC4–PC7) и *схемы управления группой B*, вырабатывающей сигналы управления *каналом B* и младшими разрядами *канала C* (PC0–PC3). Кроме этого, к *блоку управления каналами* относится *регистр управляющего слова* (PUC), в который программным путем через *буфер данных* и *внутреннюю магистраль* заносится *управляющее слово* (UC), задающее режимы работы каналов микросхемы.

## 1.2. ПОДКЛЮЧЕНИЕ ППА К МАГИСТРАЛЯМ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ



**Рис. 2.** Схема подключения ППА к магистралям микропроцессорной системы

Условное графическое изображение параллельного периферийного адаптера и схема подключения к магистралям микропроцессорной системы показаны на рисунке 2.

Назначение входных, выходных и управляющих сигналов ППА указано при описании выводов микросхемы в таблице 2.

Через выходы **D0÷D7** буфера канала данных ППА подключается к одноименным линиям магистрали данных непосредственно или через буферный элемент типа K580BA86 в том случае, если нагрузочной способности буфера канала данных недостаточно для работы с шиной данных системы. Входы **A0** и **A1**, как правило, подключаются к младшим разрядам магистрали адреса (MA), а поступивший на них код задает следующий порядок подключения буфера канала данных: 00 – к каналу 0, 01 – к каналу 1, 10 – каналу 2, 11 – к регистру управляющего слова. Код на остальных адресных линиях определяет выбранную микросхему.

Таблица 2. Функциональное назначение выводов ППА

| Вывод         | Обозначение         | Тип вывода       | Функциональное назначение выводов |
|---------------|---------------------|------------------|-----------------------------------|
| 1—4,<br>37—40 | PA3—PA0,<br>PA7—PA4 | Входы/<br>выходы | Информационный канал <i>A</i>     |
| 5             | $\overline{RD}$     | Вход             | Чтение информации                 |
| 6             | $\overline{CS}$     | Вход             | Выбор микросхемы                  |
| 7             | GND                 | —                | Общий                             |
| 9, 8          | A0, A1              | Вход             | Младшие разряды адреса            |
| 10—17         | PC7—PC4,<br>PC0—PC3 | Входы/<br>выходы | Информационный канал <i>C</i>     |
| 18—25         | PB0—PB7             | Входы/<br>выходы | Информационный канал <i>B</i>     |
| 26            | Ucc                 | —                | Напряжение питания<br>+5 В ±5%    |
| 27—34         | D7—D0               | Входы/<br>выходы | Канал данных                      |
| 35            | RES                 | Вход             | Установка в исходное состояние    |
| 36            | $\overline{WR}$     | Вход             | Запись информации                 |

Сигнал выборки  $\overline{CS0}$  с низким активным уровнем формируется схемой дешифратора **DC**, который может быть выполнен на микросхемах комбинационной логики, компараторах кодов и дешифраторах в интегральном исполнении. В МПС, имеющих в своем составе контроллер прямого доступа к памяти (ПДП), работу дешифратора выборки следует блокировать на время циклов ПДП по активному уровню сигнала **AEN** (**A**dress **E**nable), поскольку в этот промежуток времени сигналы магистрали управления (МУ) формируются контроллером ПДП, а не центральным процессором. В МПС, построенных на МП K580BM80, дешифрация устройств ввода/вывода может быть упрощена в том случае, если их количество не превышает шести. Поскольку МП K580BM80 при обращении к внешним устройствам командами **IN** и **OUT** дублирует на линиях **A15÷A8** код, выставляемый по линиям **A7÷A0**, входы  $\overline{CS}$  микросхем внешних устройств могут подключаться непосредственно к линиям **A15÷A10** МА без дополнительной дешифрации и однозначно адресоваться кодами 111110XXb, 111101XXb, 111011XXb, 110111XXb, 101111XXb, 011111XXb.

Входы  $\overline{WR}$  и  $\overline{RD}$  определяют направление передачи данных и подключаются к линиям  $\overline{IOWR}$ ,  $\overline{IORD}$  МУ в том случае, если ППА подключен как устройство ввода/вывода, или к линиям  $\overline{MEMWR}$ ,  $\overline{MEMRD}$  – если регистры ППА включены как ячейки памяти. В простых микропроцессорных

системах, в которых пространство ячеек памяти и пространство устройств ввода/вывода не разделены, входы  $\overline{WR}$  и  $\overline{RD}$  подключаются непосредственно к соответствующим выводам микропроцессора.

При включении ППА в пространство устройств ввода/вывода программирование и обмен данными с ним осуществляется командами ассемблера IN и OUT. При работе с регистрами ППА как с ячейкам оперативного запоминающего устройства (ОЗУ), в полном объеме применимы команды обращения к памяти. В том случае, если пространство ячеек памяти и пространство устройств ввода/вывода не разделены, доступ к регистрам ППА возможен как с помощью команд обращения к памяти, так и с помощью команд ввода/вывода с учетом специфики работы последних применительно к конкретному типу процессора. Так команды IN 61H и OUT 61H МП К580ВМ80 выполняются в этом случае эквивалентно командам LDA 6161H и STA 6161H.

Вход установки БИС в исходное состояние **RES** подключается к линии начального сброса **RESET** магистрали управления МПС.

Для связи с периферийными устройствами используются 24 линии ввода/вывода, сгруппированные в три 8-разрядных канала PA, PB, PC, направление передачи информации и режимы работы которых определяются программным способом. Нагрузочная способность выводов каналов такова, что позволяет подключать к каждому выходу только один вход ТТЛ элементов, поэтому при большой суммарной нагрузке необходимо вводить специальные буферные элементы.

Сигналы управления работой ППА  $\overline{WR}$ ,  $\overline{RD}$  и  $\overline{CS}$  вместе с адресными входами A0, A1 задают вид операции, выполняемой БИС ППА (табл. 2).

**Таблица 2.** Операции, задаваемые управляющими сигналами ППА

| Операция                        | Сигналы на входах |                 |                 |    |    |
|---------------------------------|-------------------|-----------------|-----------------|----|----|
|                                 | $\overline{CS}$   | $\overline{RD}$ | $\overline{WR}$ | A1 | A0 |
| <b>Операции вывода (запись)</b> |                   |                 |                 |    |    |
| Запись управляющего слова из МП | 0                 | 1               | 0               | 1  | 1  |
| Запись в канал А                | 0                 | 1               | 0               | 0  | 0  |
| Запись в канал В                | 0                 | 1               | 0               | 0  | 1  |
| Запись в канал С                | 0                 | 1               | 0               | 1  | 0  |
| <b>Операции ввода (чтение)</b>  |                   |                 |                 |    |    |
| Чтение из канала А              | 0                 | 0               | 1               | 0  | 0  |
| Чтение из канала В              | 0                 | 0               | 1               | 0  | 1  |
| Чтение из канала С              | 0                 | 0               | 1               | 1  | 0  |
| <b>Операции блокировки</b>      |                   |                 |                 |    |    |
| Запрещенная комбинация          | 0                 | 0               | 1               | 1  | 1  |
| Отключение ППА от D7–D0         | 1                 | x               | x               | x  | x  |

**Примечание.** x - безразличное состояние сигнала



### 1.3. НАЧАЛЬНАЯ ИНИЦИАЛИЗАЦИЯ АДАПТЕРА

Группы информационных каналов ППА полностью независимы друг от друга и каждая может быть настроена в свой режим работы. Режим работы каналов РА, РВ, РС определяется содержимым *регистра управляющего слова*. Произведя запись *управляющего слова* в РУС, можно перевести группы каналов в один из трех режимов работы:

- режим 0** — простой ввод/вывод;
- режим 1** — стробируемый ввод/вывод;
- режим 2** — двунаправленный канал.

*При подаче сигнала начального сброса RESET* регистр управляющего слова устанавливается в состояние, при котором *все каналы настраиваются на работу в режиме 0 для ввода информации*, что позволяет избежать конфликтов с подключенными к БИС внешними устройствами на этапе начальной инициализации.

Режимы работы ППА определяются в процессе начальной установки записью *управляющего слова* в *регистр управляющего слова* по адресу  $A0=1, A1=1$  и программируются с помощью простых операций вывода или записи в память. *Управляющее слово* представляет собой 8-битное число, отдельные разряды которого определяют настройку каналов БИС. Формат управляющего слова приведен в таблице 3. Режимы работы каналов можно изменять как в начале, так и в процесс выполнения программы, что позволяет обслуживать различные периферийные устройства в определенном порядке одной микросхемой. При изменении режима работы любого канала все входные и выходные регистры каналов и триггеры состояния сбрасываются.

**Таблица 3. Формат управляющего слова (побитно) для выбора режима работы**

| D7                         | D6   | D5 | D4                                   | D3                                   | D2                             | D1                                   | D0                                   |
|----------------------------|--|----|--------------------------------------|--------------------------------------|--------------------------------|--------------------------------------|--------------------------------------|
| Признак управляющего слова | Режим работы каналов группы А:               |    | Направление передачи канала РА0-РА7: | Направление передачи канала РС4-РС7: | Режим работы каналов группы В: | Направление передачи канала РВ0-РВ7: | Направление передачи канала РС0-РС3: |
| 1 – установка режимов      | 00 – режим 0<br>01 – режим 1<br>1x – режим 2 |    | 0 – вывод;<br>1 – ввод               | 0 – вывод;<br>1 – ввод               | 0 – режим 0;<br>1 – режим 1    | 0 – вывод;<br>1 – ввод               | 0 – вывод;<br>1 – ввод               |

**Примечание.** x - безразличное состояние сигнала

В дополнение к основным режимам работы микросхема обеспечивает возможность программной независимой установки в "1" и сброса в "0" любого из разрядов регистра канала РС. Формат *управляющего слова* установки/сброса отдельных разрядов регистра канала РС приведен в таблице 4.

**Таблица 4.** Формат УС установки и сброса разрядов регистра канала РС

| D7  | D6            | D5 | D4 | D3  | D2             | D1             | D0  |
|---|---------------|----|----|---|----------------|----------------|---|
| Признак управляющего слова                      | Не определено |    |    | Номер устанавливаемого разряда канала РС: |                |                | Значение устанавливаемого разряда:            |
| 0 – установка/сброс разрядов регистра канала РС | xxx           |    |    | 000 – разряд 0                            | 001 – разряд 1 | 010 – разряд 2 | 011 – разряд 3                                |
|   |               |    |    | 100 – разряд 4                            | 101 – разряд 5 | 110 – разряд 6 | 111 – разряд 7                                |
|   |               |    |    |   |                |                | 0 – установить в "0";<br>1 – установить в "1" |

**Примечание.** x - безразличное состояние сигнала

Сброс/установку разрядов канала РС можно использовать для выработки сигналов запроса прерывания от ППА. Эта особенность микросхемы позволяет программно запрещать или разрешать обслуживание любого внешнего устройства ввода/вывода без анализа запроса прерывания в схеме прерываний системы.

#### 1.4. РЕЖИМЫ РАБОТЫ АДАПТЕРА

Группы каналов БИС ППА неравноправны в выборе режима работы. Так канал А может работать в любом из трех режимов, канал В – в режимах 0 и 1. Канал С может быть использован для передачи данных только в режиме 0, а в остальных режимах он служит для передачи управляющих сигналов, сопровождающих процесс обмена по каналам А и В.

Основные режимы работы микросхемы предоставляют следующие возможности обмена с внешними устройствами.

**В режиме 0** обеспечивается возможность синхронной и асинхронной программно управляемой передачи данных через два независимых 8-разрядных канала РА и РВ и два 4-разрядных канала РС.

**В режиме 1** обеспечивается возможность вывода (или ввода) информации из (или в) периферийного устройства через два независимых 8-разрядных канала РА и РВ по сигналам квитирования. При этом линии канала РС используются для приема и выдачи сигналов управления обменом.

**В режиме 2** обеспечивается возможность обмена информацией с периферийными устройствами через двунаправленный 8-разрядный канал РА по сигналам квитирования. Для передачи и приема сигналов управления обменом используются пять линий канала РС.

Рассмотрим специфику работы каналов в основных режимах более подробно.

**Режим 0** применяется при синхронном обмене или при программной организации асинхронного обмена. При работе микросхемы в режиме 0 обеспечивается простой ввод или вывод информации через любой из трех каналов, причем сигналов управления обменом информацией с периферийным устройством не требуется. В этом режиме микросхема представляет собой совокупность двух 8-разрядных и двух 4-разрядных каналов ввода/вывода, причем выводимые данные фиксируются в буферных регистрах портов по сигналу  $\overline{WR}$ , а вводимые из ВУ — не запоминаются и на МД передается текущее значение входных данных при действии нулевого уровня сигнала  $\overline{RD}$ . В режиме 0 возможны 16 различных комбинаций схем ввода/вывода каналов РА, РВ, РС, которые приведены в таблице 5.

**Таблица 5. Комбинации схем ввода/вывода каналов ППА в режиме 0**

| УПРАВЛЯЮЩЕЕ СЛОВО (HEX) | Направление передачи информации |                         |                   |                         |
|-------------------------|---------------------------------|-------------------------|-------------------|-------------------------|
|                         | КАНАЛ А (РА0–РА7)               | КАНАЛ С (СТ.) (РС4–РС7) | КАНАЛ В (РВ0–РВ7) | КАНАЛ С (МЛ.) (РС0–РС3) |
| 80                      | Вывод                           | Вывод                   | Вывод             | Вывод                   |
| 81                      | Вывод                           | Вывод                   | Вывод             | <i>Ввод</i>             |
| 82                      | Вывод                           | Вывод                   | <i>Ввод</i>       | Вывод                   |
| 83                      | Вывод                           | Вывод                   | <i>Ввод</i>       | <i>Ввод</i>             |
| 88                      | Вывод                           | <i>Ввод</i>             | Вывод             | Вывод                   |
| 89                      | Вывод                           | <i>Ввод</i>             | Вывод             | <i>Ввод</i>             |
| 8A                      | Вывод                           | <i>Ввод</i>             | <i>Ввод</i>       | Вывод                   |
| 8B                      | Вывод                           | <i>Ввод</i>             | <i>Ввод</i>       | <i>Ввод</i>             |
| 90                      | <i>Ввод</i>                     | Вывод                   | Вывод             | Вывод                   |
| 91                      | <i>Ввод</i>                     | Вывод                   | Вывод             | <i>Ввод</i>             |
| 92                      | <i>Ввод</i>                     | Вывод                   | <i>Ввод</i>       | Вывод                   |
| 93                      | <i>Ввод</i>                     | Вывод                   | <i>Ввод</i>       | <i>Ввод</i>             |
| 98                      | <i>Ввод</i>                     | <i>Ввод</i>             | Вывод             | Вывод                   |
| 99                      | <i>Ввод</i>                     | <i>Ввод</i>             | Вывод             | <i>Ввод</i>             |
| 9A                      | <i>Ввод</i>                     | <i>Ввод</i>             | <i>Ввод</i>       | Вывод                   |
| 9B                      | <i>Ввод</i>                     | <i>Ввод</i>             | <i>Ввод</i>       | <i>Ввод</i>             |

**Режим 1** обеспечивает стробируемый однонаправленный обмен информацией с внешним устройством. В режиме 1 передача данных осуществляется только через каналы PA и PB, а линии канала PC используются для приема и выдачи сигналов управления обменом (сигналов квитирования).

Если один из каналов запрограммировать в режим 1, то остальные 13 интерфейсных линий можно использовать в режиме 0. Если оба канала запрограммированы в режим 1, то оставшиеся 2 интерфейсные линии канала PC могут быть настроены на ввод или вывод.

Работу канала в этом режиме сопровождают три управляющих сигнала. Для ввода информации в режиме 1 управляющие сигналы следующие:

- строб приема (STB) – входной сигнал, формируемый внешним устройством указывает на готовность ВУ к вводу информации;
- подтверждение приема (IBF) – выходной сигнал ППА, сообщающий ВУ об окончании приема данных в канал; формируется по спаду STB;
- запрос прерывания (INTR) – выходной сигнал ППА, информирующий МП о завершении приема информации в канал; высокий уровень сигнала устанавливается при STB=1, IBF=1 и INTE=1; сбрасывается спадом сигнала  $\overline{RD}$ .

Для операции ввода управление сигналом разрешения прерывания INTE канала A осуществляется по линии PC4, а канала B – по линии PC2.

Для вывода информации в режиме 1 используются следующие управляющие сигналы:

- строб записи (OBF) – выходной сигнал, указывающий ВУ о готовности к выводу; формируется по фронту сигнала  $\overline{WR}$ ;
- подтверждение записи (ACK) – входной сигнал от ВУ, подтверждающий прием информации из ППА;
- запрос прерывания (INTR) – выходной сигнал ППА, информирующий МП о завершении операции вывода информации; высокий уровень сигнала устанавливается по фронту сигнала ACK при OBF=1 и INTR=1; сбрасывается сигналом  $\overline{WR}$ .

Для операции вывода управление сигналом разрешения прерывания INTE канала A осуществляется по линии PC6, а канала B – по линии PC2.

Последовательность формирования управляющих сигналов по линиям порта C в зависимости от сигналов ВУ и системных сигналов  $\overline{RD}$  и  $\overline{WR}$  показана на рисунке 3. Рассмотрим эту последовательность на примере работы группы A, соответствующие разряды группы B будут указаны в скобках.

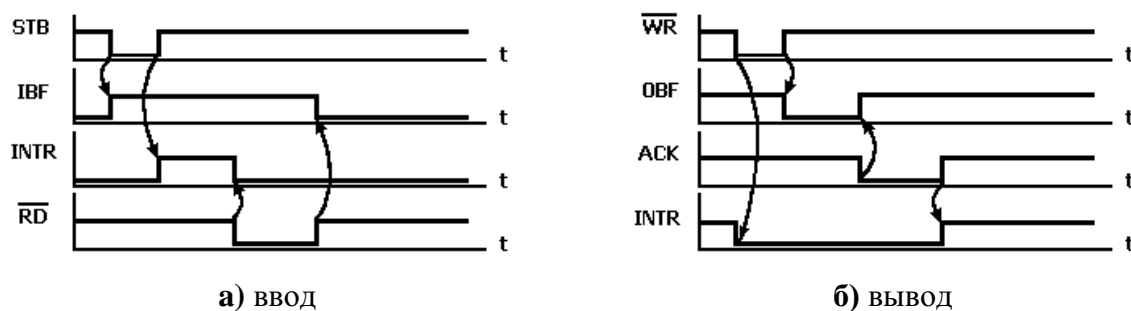
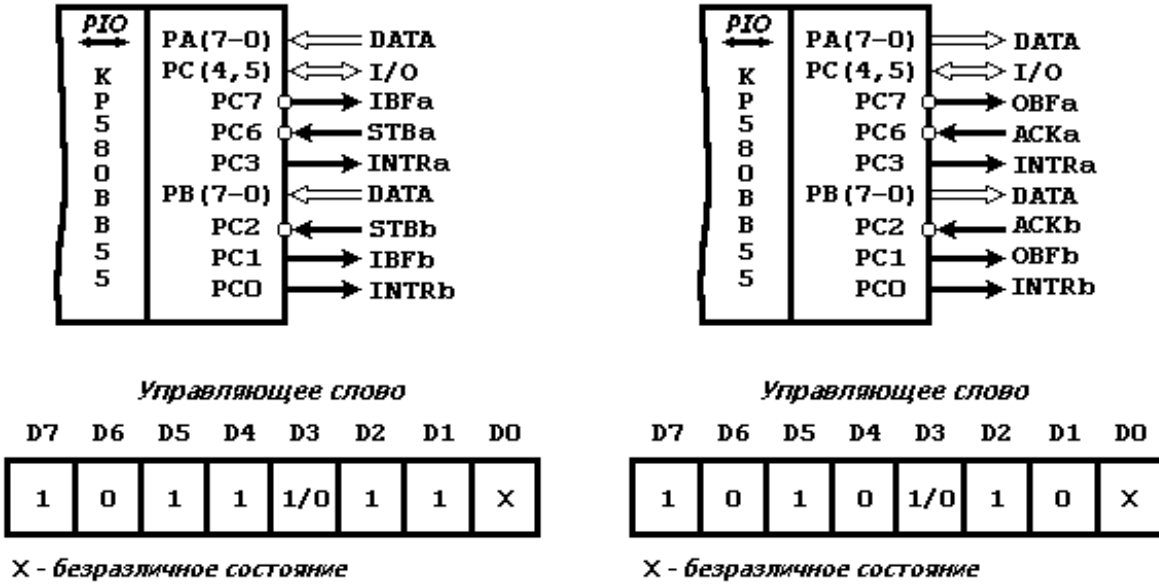


Рис. 3. Диаграммы сигналов управления в режиме 1

При вводе информации ВУ формирует сигнал STB, низкий уровень которого, поступая на вход PC4 (PC2), обеспечивает запись информации в буферный регистр порта данных. Одновременно с началом записи на выходе PC5 (PC1) формируется высокий уровень сигнала IBF, информирующего ВУ о том, что идет цикл ввода. По окончании инверсного импульса STB при наличии программного разрешения, т. е. предварительной установки разряда PC4 (PC2), на выходе PC3 (PC0) формируется сигнал INTR запроса прерывания, указывающий, что МП может вводить данные, поступившие в порт из ВУ. Формирование сигнала INTR можно запретить программным сбросом разряда PC4 (PC2) порта С. Сигнал INTR используется для перехода к подпрограмме ввода данных через соответствующий порт. Этот сигнал сбрасывается в момент прихода импульса  $\overline{RD}$ . Фронт сигнала  $\overline{RD}$  устанавливает низкий уровень сигнала IBF, извещая ВУ о возможности ввода следующего байта данных.

При выводе информации (рис. 3, б) цикл обмена начинает МПС, подавая на БИС параллельного ввода/вывода сигнал  $\overline{WR}$ . Если разряд PC6 (PC2) порта С был предварительно установлен, то о том, что идет обмен, извещает низкий уровень сигнала INTR на выходе PC3 (PC0). Этот сигнал сбрасывается по спаду сигнала  $\overline{WR}$ , фронт которого, в свою очередь, сбрасывает сигнал OBF на выходе PC7 (PC1) указывая ВУ, что данные введены в буфер порта. Данные считываются по сигналу ACK подтверждения чтения, подаваемому ВУ на вход PC6 (PC2). В момент окончания считывания (по фронту ACK) устанавливается высокий уровень сигнала INTR, давая знать МП, что ВУ приняло данные и можно начинать следующий цикл обмена.

На рисунке 4, а приведен пример конфигурации ППА в режиме 1 и соответствующее ему управляющее слово для ввода по каналам А, В, а на рис. 4, б — для вывода. Не используемые для передачи управляющих сигналов линии PC7, PC6 (рис. 4 а) и PC5, PC4 (рис. 4 б) могут быть запрограммированы на ввод ( $D3=1$ ) или вывод ( $D3=0$ ).



а) ввод по каналам А, В

б) вывод по каналам А, В

Рис. 4. Конфигурация ППА в режиме 1 и соответствующее управляющее слово

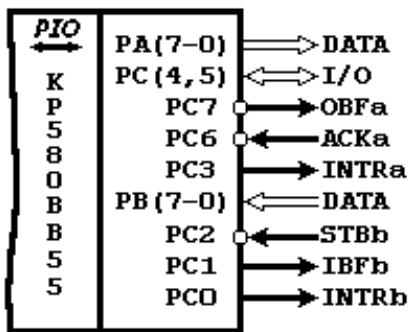


Рис. 5. Конфигурация ППА в режиме 1 для вывода информации по каналу А и вводу по каналу В.

На рисунке 5 изображен вариант конфигурации ППА в режиме 1 для вывода информации по каналу А и вводу по каналу В. Управляющее слово этого варианта имеет вид **1 0 1 0 D3 1 1 Xb**, где D3 определяет работу линий PC5, PC4 на ввод или вывод.

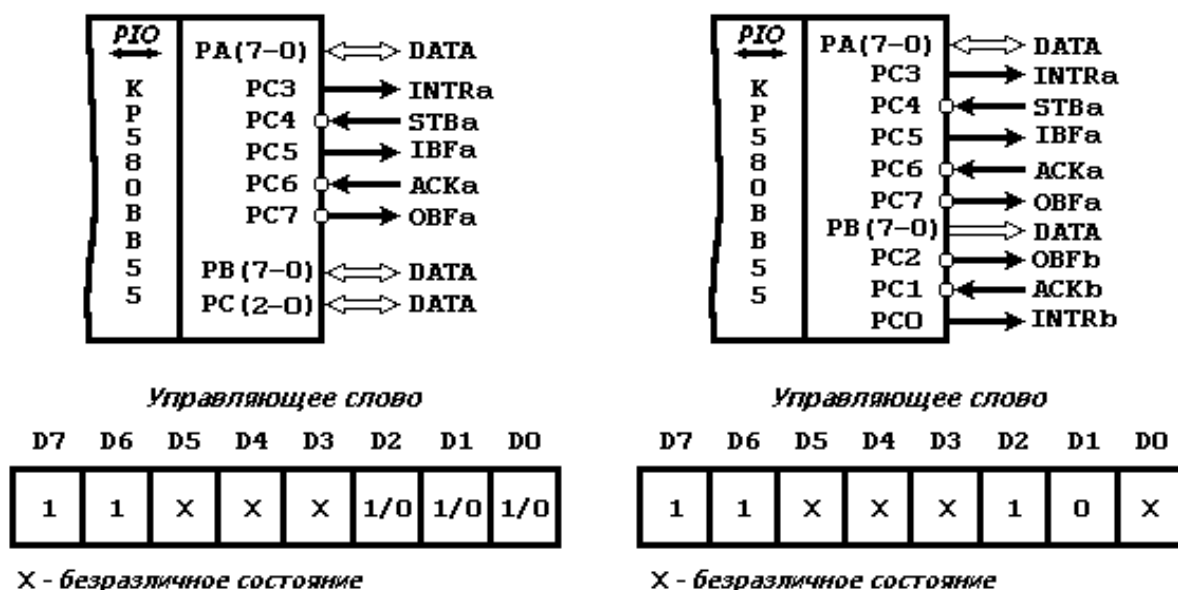
Установка/сброс триггера разрешения прерывания каждого из каналов А и В, осуществляется управляющим словом установки/сброса определенного разряда канала С.

Если триггер разрешения прерывания соответствующего канала установлен (INTE=1), то ППА может сформировать сигнал запроса прерывания при готовности внешнего устройства к вводу или выводу.

**Режим 2** обеспечивает двунаправленную асинхронную передачу информации по каналу А к ВУ и обратно. Процесс обмена сопровождаются 5 управляющих сигналов, подаваемых по линиям PC7–PC3. Оставшиеся 11 интерфейсных линий могут настраиваться в режим 0 или режим 1. Функции сигналов управления, используемых при передаче информации в режиме 2, и

временные соотношения между ними такие же, как и в режиме 1. Управление установкой внутреннего сигнала INTE для операции ввода осуществляется по линии PC4, а для операции вывода – по линии PC6. Сигнал INTR формируется в зависимости от предварительной установки двух разрядов PC6 и PC4 порта C одновременно.

Распределение сигналов по интерфейсным линиям и управляющее слово режима 2 приведены на рис. 6, а. Разряд D0 в этой конфигурации ППА определяет настройку на ввод ("1") или вывод ("0") интерфейсных линий PC2, PC1 и PC0. Остальные разряды соответствуют формату управляющего слова установки режимов работы (табл. 3).



а) канал А – двунаправленный обмен;  
каналы В и C2–C0 – обмен в режиме 0

б) канал А – двунаправленный обмен;  
канал В – вывод в режиме 1

**Рис. 6.** Конфигурация ППА в режиме 2 и соответствующее управляющее слово

На рис. 6, б показан один из возможных вариантов комбинированного режима работы ППА, в котором канал А запрограммирован в режим 2, а канал В — на вывод в режиме 1.

Если микросхема запрограммирована для работы в режиме 1 или 2, то состояние каждого сигнала управления в протоколе установления связи с периферийным устройством, принимаемого и выдаваемого через выходы канала ВС, фиксируется в регистре канала ВС. Это позволяет простым чтением содержимого регистра канала С проверить состояние каждого периферийного устройства, подключенного к микросхеме, и в соответствии с

состоянием внешнего устройства оперативно изменять процесс прохождения программы.

Для чтения информации состояния используется обычная операция чтения канала С. Форматы слова состояния для режимов 1 и 2 показаны на рисунке 7, а–в.

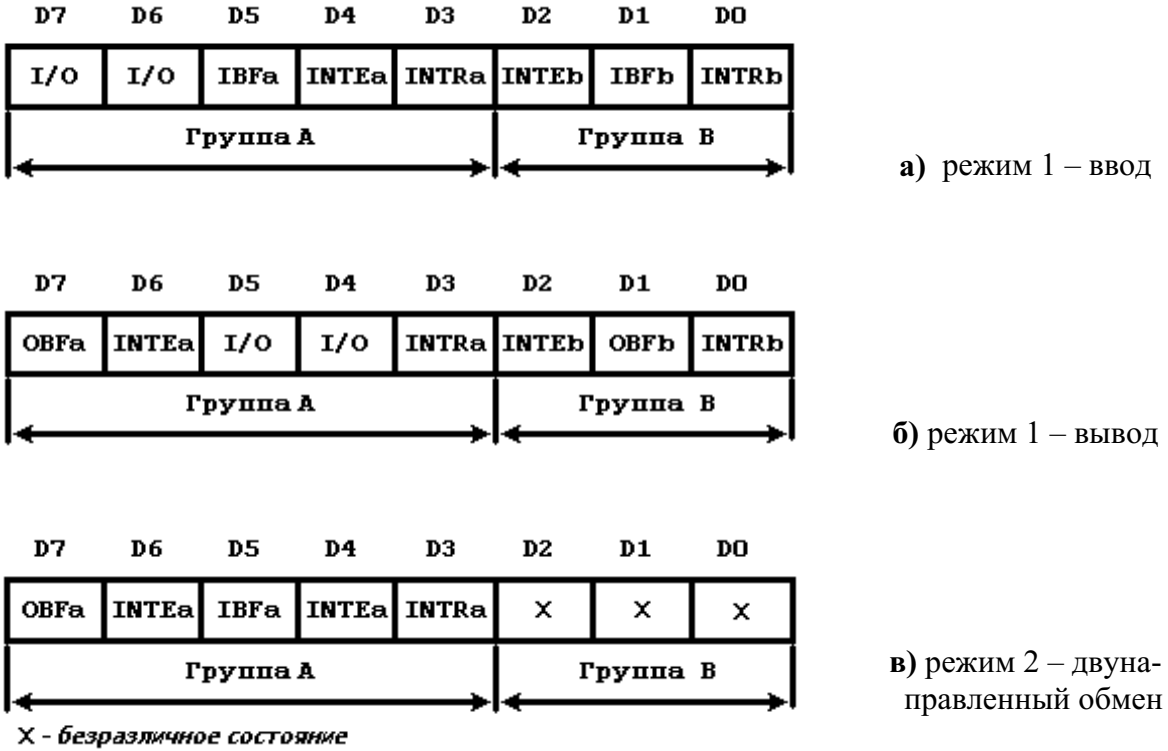


Рис. 7. Форматы слова состояния для режимов 1 и 2

## 1.5. ОСНОВЫ ПРОГРАММИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ НА ЯЗЫКЕ АССЕМБЛЕРА

Ассемблерные программы обладают преимуществом компактности кода и максимальным быстродействием, вследствие этого они чаще всего используются при работе микропроцессорных систем в режиме реального времени, а также при необходимости управления внешними устройствами или обмена данными с ними. Операционные системы МПС обычно содержат низкоуровневые подпрограммы (драйверы) для работы со стандартными устройствами ввода/вывода. Потребность в написании ассемблерной подпрограммы может быть вызвана заменой стандартного устройства, оптимизацией кода существующих драйверов, необходимостью обслуживания нестандартной периферии.



Рассмотрим в качестве примера задачу управления через каналы ППА матричным принтером по упрощенному протоколу интерфейса Centronics и выведем блок кодов ASCII заданного размера на печать.

### 1.5.1. Формулировка задачи

Первым этапом решения задачи является составление алгоритма, включающего в себя как словесное описание, так и любые виды формализации — таблицы, формулы, графики, принципиальные и блок-схемы. Этап формулирования задачи является одним из самых важных, поскольку ошибки, допущенные в процессе составления алгоритма, сводят на нет все дальнейшие усилия по составлению программы и кодированию.

Понятие Centronics относится к набору сигналов, протоколу обмена и разъему, устанавливаемому в принтерах. Большинство современных принтеров совместимы с интерфейсом Centronics. Временная диаграмма сигналов обмена данными по упрощенному протоколу Centronics приведена на рисунке 8.



Рис. 8. Временная диаграмма сигналов обмена данными по упрощенному протоколу Centronics

Передача при минимальном наборе управляющих сигналов начинается с проверки источником сигнала **BUSY**. Если он равен "0", то источник приступает к передаче байта данных. Для передачи байта источник выставляет на линии D0–D7 байт данных и выдает сигнал **STROBE**.

Приемник по сигналу **STROBE** читает данные с шины данных и выставляет сигнал **BUSY** на время его обработки. По окончании обработки приемник снимает сигнал **BUSY**. Если после приема байта приемник по какой-либо причине не готов принимать данные, то он не снимает сигнал **BUSY**. При программной реализации обмена по указанному протоколу желательно ограничить время ожидания снятия **BUSY** (обычно 30 – 45 сек), иначе возможно зависание программы.

Большинство задач допускает альтернативные равносильные решения. На этапе формулировки задачи следует провести их оценку и определить возможность и целесообразность практической реализации. Так в рассматриваемой задаче аппаратные средства оговорены заранее и изменение их не предусмотрено. С точки зрения программного воплощения алгоритма обмена представляется приемлемым использование режима 1 для реализации протокола. Но применение этого режима требует наличия в системе контроллера прерываний и неиспользуемой линии запроса, что никак не оговорено при обсуждении аппаратных средств задачи. Следовательно,

асинхронный программный обмен данными и сигналами управления по информационным каналам ППА в режиме 0 представляет собой наиболее приемлемую альтернативу.

Для обеспечения совместимости программы с целью использования разработанных алгоритмов в дальнейшей работе при распределении сигналов интерфейса по информационным линиям адаптера следует придерживаться существующих стандартов. Так в компьютерах IBM PC порт по базовому адресу адаптера параллельного принтера выводит байт данных. Порт с адресом на единицу большим базового занят чтением состояния принтера и контролирует сигнал **BUSY** по линии 7. Порт управления принтера отстоит от базового адреса на 2 и формирует сигнал **STROBE** по линии 0. В соответствии со стандартом следует настроить порт А ППА на вывод в режиме 0, порт В — на ввод в режиме 0, порт С адаптера — на вывод в режиме 0. Управляющие сигналы распределить по выводам ППА согласно стандарту IBM PC.

Для работы с кабелем и входными линиями принтера следует повысить нагрузочную способность выходов ППА, подключив к ним шинные формирователи типа K580BA86.

**Определение задачи:** вывод блока кодов на печать через информационные каналы ППА по упрощенному протоколу интерфейса Centronics.

**Устройства:** принтер, клавиатура.

**Датчики и сигналы:** матрица клавиш, опрашиваемая через порт клавиатуры; линии интерфейса Centronics, программно опрашиваемые и обслуживаемые информационными каналами ППА.

**Суть задачи:** при неактивном сигнале **BUSY** интерфейса принтера (порт В – линия 7), вывести байт данных в порт А (линии 0–7), подтвердить принтеру их истинность активным уровнем сигнала **STROBE** (линия PC0), по активному уровню сигнала **BUSY** снять сигнал **STROBE**. Дождаться окончания сигнала **BUSY**. Повторит процедуру, если не достигнут конец блока кодов. Обеспечить выход из процедуры по нажатию клавиши [ESC] матрицы клавиатуры.

Задача поставлена. Она содержит работу с внешними устройствами в реальном времени, программирование периферии и портов ввода/вывода.

### 1.5.2. Решение задачи

Следует определить для себя ответы на ряд вопросов следующего характера:

- а) какие данные вводятся или выводятся;
- б) какие порты используются и в какие режимы их запрограммировать;
- в) в каком виде осуществить диалог с оператором (если диалог нужен);
- г) какая информация и в какой форме необходима для вывода;
- д) возможное развитие задачи.

Возможные ответы: состояние линий интерфейса принтера преобразуется в установку соответствующих битов параллельного порта. Начало блока данных определяет регистровая пара [HL], конец блока данных определяет регистровая пара [DE]. Если интерфейс реализован на БИС K580BB55A, осуществим следующую настройку:

канал C0–C3 – вывод;  
 канал B0–B7 – ввод;  
 каналы B, C0–C3 – режим 0;  
 канал C4–C7 – вывод;  
 канал A0–A7 – вывод;  
 каналы A, C4–C7 – режим 0.

Для опроса клавиатуры с целью упрощения будем использовать стандартную подпрограмму опроса консоли, возвращающую код клавиши в случае её нажатия и код 0FFH – в противном случае (Клавиатура может опрашиваться разными программами, поэтому лучше не использовать опрос матрицы клавиш своей подпрограммой для совместимости). В качестве диалога с оператором или пользователем можно предусмотреть вывод на экран сообщения типа: "Для прекращения печати нажмите клавишу [ESC]" ("Press ESC to terminate..."), но поскольку мы пишем процедуру низкого уровня, а ESC – стандартная клавиша для отмены процесса, подобное предупреждение обычно выводится вызывающей программой. В качестве развития задачи можно предусмотреть подачу звукового сигнала, если принтер занят свыше определенного времени, что обычно свидетельствует об ошибке печати.

### 1.5.3 Проектирование программы

Существует множество различных методов проектирования программ. Метод "Сверху вниз" или "Нисходящее проектирование" является одним из наиболее простых и понятных. Суть метода заключается в том, что решаемая задача последовательно разбивается на части, каждая из которых – на более мелкие (называемые модулями). Разбиение прекращается, когда содержание модуля не вызывает видимых затруднений для программной реализации. Вполне вероятно, что отдельные модули могут в дальнейшем потребовать дополнительно и более мелкого разбиения.

Детализация разбиения и размер модулей величина субъективная, но предпочтительно, чтобы модули были небольшими, что связано с особенностями зрительного восприятия человека, и, по возможности – независимыми, для оперативной замены при внесении изменений в программу.

### 1.5.4 Проектирование модулей

При правильном проектировании программы простота функций и логики работы каждого модуля позволяют сразу же записать код на языке ассемблера. В противном случае можно описать алгоритм работы модуля на "псевдокоде" (то есть, на языке, близком к естественному). Указатель ком-

ментария “ ; ” позволяет реализовать такой подход в программах на ассемблере.

Правила “хорошего тона” в программировании предъявляют к оформлению и написанию отдельного модуля следующие требования:

- а) наличие заголовка, включающего имя вызова модуля, краткое описание выполняемой функции, а также входных и выходных данных;
- б) небольшой размер (для отладки в экранном режиме – не более 40 строк);
- в) возвращать управление по месту вызова;
- г) наличие одного входа и одного выхода;
- д) возможность обращения к другим модулям (универсальность отдельных модулей);
- е) по возможности меньшая взаимная зависимость (для оперативной замены);
- ж) наличие необходимых комментариев по тексту программы.

Выполнение последнего требования безусловно необходимо при коммерческой разработке программ и передаче программного обеспечения заказчику. Тем не менее, выполнение этого требования облегчает работу и с личными программами, позволяя всякий раз не затрачивать время на чтение кода и выяснение алгоритмов.

Перечисленные требования несут в себе некоторые элементы структурного программирования и являются в большей мере желательными, но не обязательными. Программа, написанная без учета данных требований, может быть вполне работоспособна, но трудоёмка в сопровождении и отладке.

### 1.5.5. Логическая структура ассемблерной программы

Элементы подпрограммы на языке ассемблера легче понять, если они расположены в той последовательности, в которой встречаются. Логической структуру ассемблерной программы в наиболее общем случае удобно представить в виде пяти вложенных в друг друга частей:

**Уровень 1:** общая надстройка ассемблера.

**Уровень 2:** надстройка ассемблерной подпрограммы.

**Уровень 3:** входной код.

**Уровень 4:** получение значений параметров вызывающей программы.

**Уровень 5:** выполнение кода программы, вызов служб ROM BIOS.

**Уровень 4:** передача результатов обратно вызывающей программе.

**Уровень 3:** выходной код.

**Уровень 2:** завершение надстройки ассемблерной подпрограммы.

**Уровень 1:** завершение общей надстройки ассемблера.

Данной базовой структуры можно придерживаться в большинстве интерфейсных программ, написанных для обращения к системным службам или просто в виде стандартных подпрограмм на ассемблере, однако следует иметь в виду, что конкретное кодирование будет видоизменяться в зависимости от языка программирования, используемого в вызывающей программе.

Для программ, написанных на языке ассемблера для процессора K580BM80 общий вид программы может быть примерно следующим:

```

;+-----+
;|                ОСНОВНЫЕ ИНСТРУКЦИИ ПРОГРАММЕ АССЕМБЛЕР                |
;+-----+
;
;Col.1 _____ - Поле меток и имен переменных
; | Col.2 _____ - Поле (псевдо)оператора (директивы)
; | | Col.3 _____ - Поле операндов
; | | | Col.4 _____ - Поле комментариев
;+---+ +---+ +---+ +-----+
;      ORG 100H
;      | +---+                256 (100H или другое количество
;      | +-----+           во байт) резервируются перед
;      |                   началом программы...
;      +-----+           этой директивой, она же связы-
;                           вает начало программы с абсо-
;                           лютным адресом 100H
;+-----+
;|                Начало основной процедуры                |
;+-----+
;
begin:
; |                   Метка (произвольна), сообщает
; +-----+           ассемблеру о начале выполнения
;                   программы с данного адреса или
;                   просто фиксирует начальный ад-
;                   рес (может отсутствовать).
;
;+-----+ Тело основной процедуры -----+
;|
;   ...
;|
;+-----+
;+-----+
;|                Объявление и инициализация данных                |
;+-----+
;      DB 32H, 5BH; (Размер: 1 байт, байты)
;      DW 0FFE1H; (Размер: 1 слово)
;      | _____ псевдооператор резервирования
;
;+-----+

```

```

;----- Внешние метки и константы -----
;
;      +----- псевдооператор присваивания
;      |
SYST:EQU 0F800H;_____ выход в систему
KLAV:EQU 0F81BH;_____ опрос клавиатуры без остановки программы
PRTA:EQU 0FFE4H;_____ адрес порта А ППА
PRTB:EQU 0FFE5H;_____ адрес порта В ППА
PRTC:EQU 0FFE6H;_____ адрес порта С ППА
RGYS:EQU 0FFE7H;_____ адрес регистра управляющего слова ППА

;+-----+
;|                               Секция конца программы                               |
;+-----+
;
;      END
;      |
;      +----- Директива конца основной
;                               процедуры

```

Оформим рассматриваемый нами алгоритм печати блока кодов на псевдокоде с элементами ассемблерной программы:

```

;----- Вывод блока кодов на печать -----
; [HL] - начало блока, [DE] - конец блока, [A] - выводимый байт
; Порт А (0FFE4h) биты 0-7 - данные D0-D7 (вывод).
; Порт В (0FFE5h) бит 7 - Busy (ввод).
; Порт С (0FFE6h) бит 0 - /Strobe (вывод).
; Регистр управляющего слова (0FFE7h) - порты в режим 0.
; Управляющее слово: 1000.0010b = 82h;
;                               порт C0-C3 - вывод, порт B0-B7 - ввод;
;                               порты В, C0-C3 - режим 0;
;                               порт C4-C7 - вывод, порт A0-A7 - вывод;
;                               порты А, C4-C7 - режим 0.

```

```

ORG 0D800H;_____ укажем транслятору начальный адрес

```

```

CALL INIT;_____ осуществим начальную настройку ППА

```

```

;----- Программа вывода блока данных [HL]-[DE] на печать -----
BLOCK:

```

```

LXI H,START;_____ занесем в [HL] адрес начала блока.

```

```

LXI D,FINSH;_____ занесем в [DE] адрес конца блока.

```

```

CYCLE:      ;_____ цикл вывода байтов на печать

```

```

CALL CONS;_____ проверим клавиатуру на нажатие [Esc]

```

```

JZ SYST;_____ если нажата - отмена, выход в систему

```

```

CALL TEST;_____ проверим [HL] на равенство [DE]

```

```

JC SYST;_____ если [HL]>[DE], блок кончился - в систему

```

```

CALL BOUT;_____ вывод на печать очередного байта

```

```

INX H;_____ укажем следующий байт

```

```

JMP CYCLE;_____ возврат в цикл печати блока кодов

```

```

;----- Подпрограмма вывода байта [A] на печать -----
;-- опрос клавиатуры проводить для выхода из цикла при ошибке -
BOUT:
;   ... ;_____ опрос клавиатуры на нажатие [Esc],
;   ... ;_____ если нажата [Esc] - прервать печать
;   ... ;_____ опрос линии Busy
;   ... ;_____ если Busy активен - переход к BOUT
;   ... ;_____ выдать байт данных на линии PA0-PA7
;   ... ;_____ активизировать /Strobe: "0" - в PC0
WAIT1:
;   ... ;_____ опрос клавиатуры на нажатие [Esc],
;   ... ;_____ если нажата [Esc] - прервать печать
;   ... ;_____ опрос линии Busy, не активна -> WAIT1
;   ... ;_____ снять сигнал /Strobe: "1" - в PC0
WAIT2:
;   ... ;_____ опрос клавиатуры на нажатие [Esc],
;   ... ;_____ если нажата [Esc] - прервать печать
;   ... ;_____ опрос линии Busy, активна -> WAIT2,
;   RET;_____ не активна - возврат из подпрограммы

;----- Подпрограмма проверки достижения конца блока -----
TEST:
;   ...           сравнить содержимое [HL] и [DE],
;   RET;_____ если конец блока не достигнут, то
;               возврат с признаком C=0 (NC)

;----- Подпрограмма опроса консоли - Z=1 - нажато [Esc] -----
CONS:
;   CALL KLAV;_____ опросить клавиатуру (код клавиши - в A)
;   ...           сравнить код с 1BH
;   RET;_____ если нажата то выход с признаком Z=1

;----- Подпрограмма настройки ППА в заданный режим -----
INIT:
;   MVI A,82H;_____ байт управляющего слова
;   STA RGYS;_____ заносим в регистра управляющего слова
;   ...           выставим начальное состояние /Strobe
;   RET;_____ вернемся из подпрограммы.

;----- Внешние метки и константы -----
START:EQU 1100H;_____ начало блока данных
FINSH:EQU 17FFH;_____ конец блока данных
SYST:EQU 0F800H;_____ выход в систему
KLAV:EQU 0F81BH;_____ опрос консоли без остановки программы
PRTA:EQU 0FFE4H;_____ адрес порта А ППА
PRTB:EQU 0FFE5H;_____ адрес порта В ППА
PRTC:EQU 0FFE6H;_____ адрес порта С ППА
RGYS:EQU 0FFE7H;_____ адрес регистра управляющего слова ППА
;   END;_____ укажем транслятору конец программы.
;-----

```

При переводе всего алгоритма на язык ассемблера строки псевдокода автоматически становятся комментариями.

```

;----- Вывод блока кодов на печать -----
; [HL] - начало блока, [DE] - конец блока, [A] - выводимый байт
; Порт А (0FFE4h) биты 0-7 - данные D0-D7 (вывод).
; Порт В (0FFE5h) бит 7 - Busy (ввод).
; Порт С (0FFE6h) бит 0 - /Strobe (вывод).
; Регистр управляющего слова (0FFE7h) - порты в режим 0.
; Управляющее слово: 1000.0010b = 82h;
;           порт C0-C3 - вывод, порт B0-B7 - ввод;
;           порты В, C0-C3 - режим 0;
;           порт C4-C7 - вывод, порт A0-A7 - вывод;
;           порты А, C4-C7 - режим 0.

ORG 0D800H;_____ укажем транслятору начальный адрес

CALL INIT;_____ осуществим начальную настройку ППА

;----- Программа вывода блока данных [HL]-[DE] на печать -----
BLOCK:
LXI H,START;_____ занесем в [HL] адрес начала блока.
LXI D,FINSH;_____ занесем в [DE] адрес конца блока.
CYCLE:      ;_____ цикл вывода байтов на печать
CALL CONS;_____ проверим клавиатуру на нажатие [Esc]
JZ SYST;_____ если нажата - отмена, выход в систему
CALL TEST;_____ проверим [HL] на равенство [DE]
JC SYST;_____ если [HL]>[DE], блок кончился - в систему
CALL BOUT;_____ вывод на печать очередного байта
JZ SYST;_____ нажата [Esc] при выводе байта на печать
INX H;_____ укажем следующий байт
JMP CYCLE;_____ возврат в цикл печати блока кодов

;----- Подпрограмма вывода байта [A] на печать -----
;-- опрос клавиатуры проводить для выхода из цикла при ошибке --
BOUT:
CALL CONS;_____ опросим консоль
RZ;_____ возврат с Z=1, если [Esc] - нажата
LDA PRTB;_____ опросим порт В, значение - в регистре А
ANI 80H;_____ выделим бит 7 - Busy
JNZ BOUT;_____ если он не = 0, то ждем этого в цикле
MOV A,M;_____ в регистре А - текущий выводимый байт
STA PRTA;_____ выдадим его в порт А
XRA A;_____ обнулим регистр А
STA PRTC;_____ выставим активный уровень /STROBE
WAIT1:
CALL CONS;_____ опрос клавиатуры на нажатие [Esc],
RZ;_____ возврат с Z=1, если [Esc] - нажата
LDA PRTB;_____ опросим порт В, значение - в регистре А
ANI 80H;_____ выделим бит 7 - Busy
JZ WAIT1;_____ если он не активен, то ждем этого в цикле

```



```

MVI A,01H;_____ снимаем активный уровень /STROBE,
STA PRTC;_____ послав "1" в бит 0 порта C
; с целью оптимизации алгоритма вывода байта на печать,
; цикл WAIT2 ожидания окончания сигнала Busy уберем из
; кода, поскольку этот цикл выполняется в начале под-
; программы BOUT при выводе следующего байта;
ORA A;_____ сбросим признак Z (Z=0 или NZ)
RET;_____ возврат из подпрограммы с Z=0

;----- Подпрограмма проверки достижения конца блока, -----
;----- сравнивает содержимое [HL] и [DE] -----
TEST:
MOV A,D;_____ старший байт конца блока
CMP H;_____ сравним со старшим байтом начала блока,
RNZ;_____ если они не равны - конец не достигнут
MOV A,E;_____ младший байт конца блока
CMP L;_____ сравним с младшим байтом начала блока,
RET;_____ если конец блока не достигнут, то
; возврат с признаком C=0 (NC)

;----- Подпрограмма опроса консоли - Z=1 - нажато [Esc] -----
CONS:
CALL KLAV;_____ опросить клавиатуру (код клавиши - в A)
CPI 1BH; _____ сравнить код с 1BH
RET;_____ если нажата то выход с признаком Z=1

;----- Подпрограмма настройки ППА в заданный режим -----
INIT:
MVI A,82H;_____ байт управляющего слова
STA RGYS;_____ заносим в регистра управляющего слова
MVI A,01H;_____ выставим начальное состояние /Strobe,
STA PRTC;_____ указав принтеру, что /Strobe не активен
RET;_____ вернемся из подпрограммы.

;----- Внешние метки и константы -----
START:EQU 1100H;_____ начало блока данных
FINSH:EQU 17FFH;_____ конец блока данных
SYST:EQU 0F800H;_____ выход в систему
KLAV:EQU 0F81BH;_____ опрос консоли без остановки программы
PRTA:EQU 0FFE4H;_____ адрес порта A ППА
PRTB:EQU 0FFE5H;_____ адрес порта B ППА
PRTC:EQU 0FFE6H;_____ адрес порта C ППА
RGYS:EQU 0FFE7H;_____ адрес регистра управляющего слова ППА
END;_____ укажем транслятору конец программы.
;-----

```

При реализации программы на платформе IBM PC, формируя сигнал **STROBE**, следует учитывать назначение остальных битов канала C при выполнении им функции порта управления принтером.

## 2. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

### 2.1. ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА

Порт параллельного интерфейса был введен в персональные компьютеры IBM для подключения принтера, откуда и произошло его название LPT–порт (**L**ine **P**rin**T**er – построчный принтер). В отдельных ранних версиях PC/XT и в некоторых отечественных IBM–совместимых персональных ЭВМ в этом качестве использовалась микросхема i8255 (отечественный аналог – КР580ВВ55). Микросхема ППА, располагавшаяся в адресах портов 60H÷63H, применялась в IBM PC/XT также в качестве системного порта, обслуживающего интерфейс клавиатуры, обеспечивающего чтение переключателей конфигурации, управление звуком и немаскируемыми прерываниями. На современных системных платах эти функции берет на себя интегральный чипсет, сохраняя при этом совместимость регистров по адресам ввода/вывода, исключая регистр управляющего слова.

В моделях IBM PC/AT порт параллельного интерфейса был интегрирован в отдельную БИС, в функции которой также входило обслуживание каналов последовательного ввода/вывода, дисководов, жесткого диска и GAME–порта. Совместимость с ППА сохранялась на уровне трёх параллельных каналов, имеющих жестко заданную настройку, позволяющую программным способом реализовать протокол передачи данных Centronics.

Практические задания, использующие настройку каналов аналогичную порту LPT, легко реализуются в среде программного эмулятора при отображении портов параллельного периферийного адаптера виртуальной машины K580BM80 на регистры БИС параллельного интерфейса материнской платы. Для изучения более сложных режимов работы ППА может использоваться специальная плата расширения. Принципиальная схема интерфейсной платы, обеспечивающей подключение к системной шине компьютера IBM PC через слот расширения стандарта ISA двух или более параллельных периферийных адаптеров КР580ВВ55, а также программируемого интервального таймера КР580ВИ53 представлена на рисунке 9.

Шинные формирователи DD2, DD3 увеличивает нагрузочную способность адресных (A0÷A11) и управляющих выводов шины ISA (**IOWR**, **IORD**, **AEN**, **RES**). На микросхемах DD5, DD9, на входы которых через шинные формирователи поданы сигналы A5÷A9 шины адреса компьютера, выполнен селектор адресов интерфейсной платы диапазона 300–31Fh, закрепленных стандартами IBM за устройствами пользователя или макетными платами. При выполнении компьютером команд чтения из портов с адресами от 300h до 31Fh или записи в эти же порты на выводе 8 DD9 фор-

мируется импульс низкого логического уровня, разрешающий работу микросхемы DD8. Дешифратор DD8, ко входам которого подключены буферизированные сигналы A2÷A4 адресной шины, разбивает адресное пространство устройств пользователя на участки по 4 байта (300H–303H ÷ 31CH–31FH), формируя на своих выходах сигналы выборки CS низкого уровня при обращении по этим адресам.

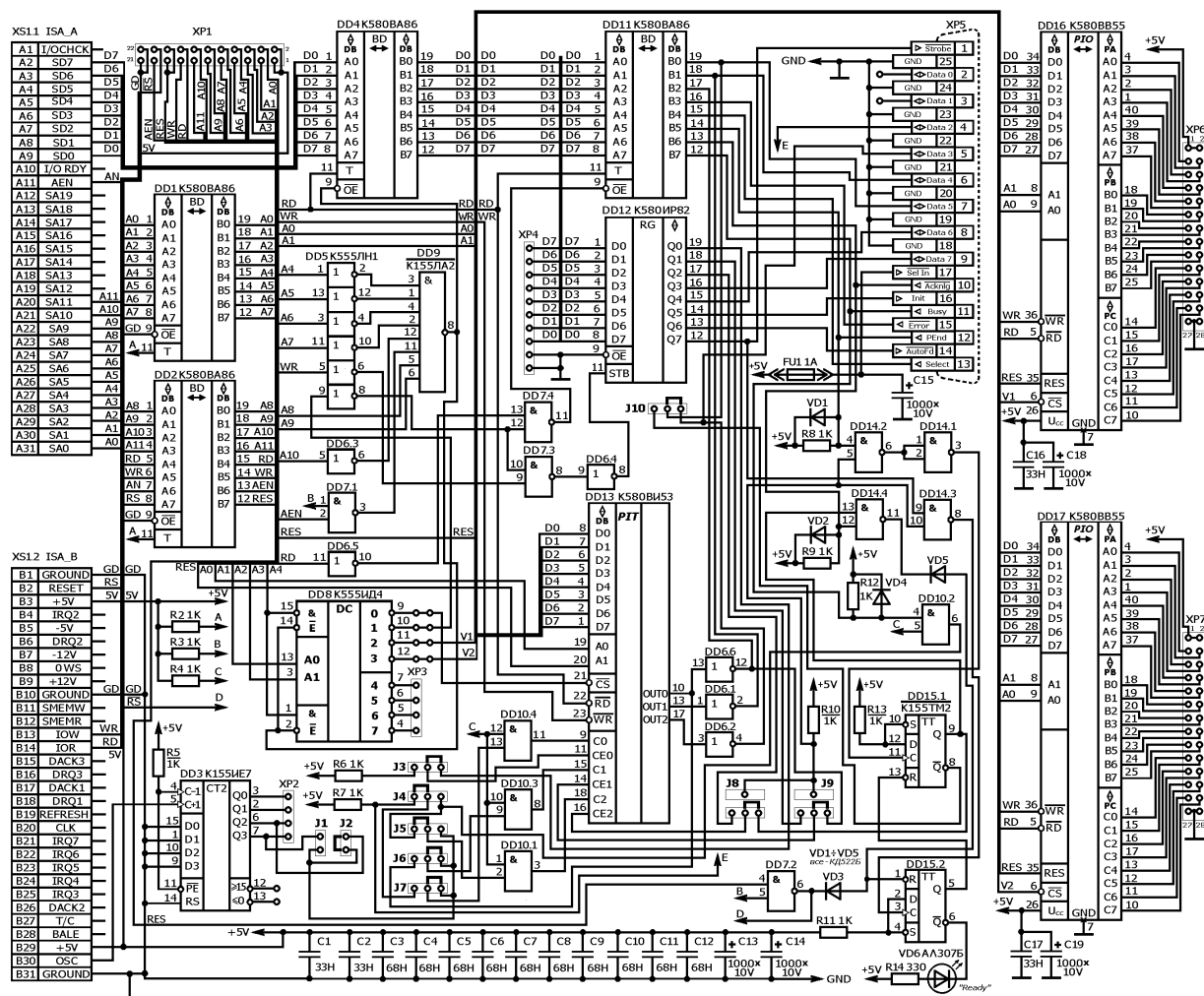


Рис. 9. Принципиальная схема платы расширения

Сигналы A0 и A1 подаются непосредственно на адресные входы программируемых БИС, осуществляя выборку внутренних регистров микросхем. Операции чтения или записи производятся по формируемым процессором компьютера сигналам  $\overline{IOWR}$  или  $\overline{IORD}$ . Для исключения сбоев на дешифратор DD5, DD9 подан сигнал AEN, блокирующий его при работе компьютера в режиме ПДП. Шинный формирователь DD4 увеличивает нагрузочную способность выводов данных (D0÷D7) программируемых БИС при работе на шину данных компьютера. Счетчик DD3 осуществляет деление тактовой частоты OSC, стандартной для всех системных плат IBM PC (14,31818 МГц), до величины, приемлемой для тактирования про-

граммируемого таймера (1,7897725 МГц – выв. 6 и 0,89488625 МГц – выв. 7). С помощью переключков J1 и J2 можно выбрать необходимую частоту тактирования.

Для соединения с внешними устройствами сигнальные цепи таймера и управляющих регистров, а также информационных каналов параллельных адаптеров DD16, DD17 выведены на заднюю панель компьютера соответственно через разъёмы XP5, XP6, XP7, закрепленные на установочных планках. Согласно приведенной принципиальной схеме (рис. 9), регистры ППА DD16, DD17 доступны для записи/чтения в пространстве ввода/вывода компьютера IBM PC, как ячейки с адресами 308H–30BH и 30CH–30FH. Адреса могут быть изменены перестановкой переключков на выводах дешифратора DD8.

Программа — эмулятор, представляющая собой кросс-средство для выполнения команд процессора K580BM80 (i8080) на платформе x86, позволяет обратиться к регистрам параллельных периферийных адаптеров DD16, DD17 как к ячейкам памяти по адресам 0FFE4h ÷ 0FFE7H, 0FFF4H ÷ 0FFF7H соответственно. В пространстве адресов внешних устройств программа эмулятора рассматривает эти ячейки как порты принтера/программатора и резервные порты.

## 2.2. ПРАКТИЧЕСКИЕ ЗАДАНИЯ

### Задание 1.1

Осуществить ввод аналоговой информации в видео-ОЗУ микропроцессорной системы методом однобитного квантования и её последующий вывод через параллельный периферийный адаптер K580BB55A.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

**регистр А - 0FFE0H;**  
**регистр В - 0FFE1H;**  
**регистр С - 0FFE2H;**  
**регистр управляющего слова - 0FFE3H.**

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH, адреса видео-ОЗУ: 9100H÷0BFFFH. При выполнении задания реализовать следующую настройку ППА: режим 0 – простой ввод/вывод,

канал А – вывод,

канал В – ввод,

канал С – вывод.

Ввод аналоговой информации осуществить через компаратор, подключенный к разряду 0 канала В, вывод — через разряды 5 и 7 канала С. Для управления использовать стандартную системную подпрограмму 0C803H, возвращающую в регистре А микропроцессора код нажатой клавиши. Использовать код 0DH (“Enter”) для управления вводом/выводом и код 1BH (“Esc”) — для выхода в системный монитор по адресу 0F800H.

Для начальной очистки видео–ОЗУ использовать стандартную системную подпрограмму 0C80FH, вызвав её дважды при наличии управляющего кода 1FH в регистре А микропроцессора.

### Задание 1.2

Осуществить ввод импульсной последовательности в видео–ОЗУ микропроцессорной системы и её последующий вывод через параллельный периферийный адаптер K580BB55A. Частота поступления импульсов меняется в диапазоне  $5 \div 500$  Гц.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

**регистр А - 0FFE0H;**  
**регистр В - 0FFE1H;**  
**регистр С - 0FFE2H;**  
**регистр управляющего слова - 0FFE3H.**

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH, адреса видео–ОЗУ: 9100H÷0BFFFH. При выполнении задания реализовать следующую настройку ППА: режим 0 – простой ввод/вывод,

канал А – вывод,  
 канал В – ввод,  
 канал С – вывод.

Ввод импульсной последовательности осуществить через разряд 0 канала В, вывод — через разряды 5 и 7 канала С. Для управления использовать стандартную системную подпрограмму 0C803H, возвращающую в регистре А микропроцессора код нажатой клавиши. Использовать код 0DH (“Enter”) для управления вводом/выводом и код 1BH (“Esc”) — для выхода в системный монитор по адресу 0F800H. Для начальной очистки видео–ОЗУ использовать стандартную системную подпрограмму 0C80FH, вызвав её дважды при наличии управляющего кода 1FH в регистре А микропроцессора. При вводе применить простейший алгоритм сжатия данных, предусматривающий хранение информации об амплитуде сигнала в старшем разряде байта, а значение длительности — в 7 младших битах.

### Задание 1.3

Осуществить вывод импульсной последовательности вида меандр (скважность 2) попеременно с частотой F и 2F через параллельный периферийный адаптер K580BB55A микропроцессорной системы.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

**регистр А - 0FFE0H;**  
**регистр В - 0FFE1H;**  
**регистр С - 0FFE2H;**  
**регистр управляющего слова - 0FFE3H.**

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ППА:  
режим 0 – простой ввод/вывод,

канал А – вывод,

канал В – ввод,

канал С – вывод.

Вывод импульсной последовательности осуществить через разряды 5 и 7 канала С, значение F выбрать произвольно из звукового диапазона частот. Тактовая частота микропроцессора – 2 МГц. Для управления использовать разряд 1 канала В ППА, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку прекращает генерацию и инициализирует выход в системный монитор по адресу 0F800H.

### Задание 1.4

Осуществить счет импульсов последовательности вида меандр (скважность – 2), используя параллельный периферийный адаптер K580BB55A микропроцессорного контроллера. Регистры ППА имеют следующие адреса в совмещенной карте памяти:

**регистр А – 0FFE0H;**

**регистр В – 0FFE1H;**

**регистр С – 0FFE2H;**

**регистр управляющего слова – 0FFE3H.**

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ППА:  
режим 0 – простой ввод/вывод,

канал А – вывод,

канал В – ввод,

канал С – вывод.

Счет импульсов осуществить через разряд 0 канала В ППА микропроцессорного контроллера. Тактовая частота процессора – 2 МГц. Для управления использовать стандартную подпрограмму опроса клавиатуры без остановки программы 0C81BH, возвращающую код клавиши в регистре А микропроцессора в случае её нажатия и код 0FFH – в противном случае. Управляющий код 0DH (“Enter”) устанавливает счетчик в исходное состояние 0000d, а код 1BH (“Esc”) инициализирует выход в системный монитор по адресу 0F800H.

Для вывода текущего состояния счетчика на дисплей использовать системную подпрограмму 0C818H, осуществляющую вывод строки символов на экран по следующему соглашению:

```
MT0:DB  1FH
MT1:DB  0CH
      DB  20H
```

MS3:DB 30H - разряд 3 (старший) счетчика,  
 MS2:DB 30H - разряд 2 счетчика,  
 MS1:DB 30H - разряд 1 счетчика,  
 MS0:DB 30H - разряд 0 (младший) счетчика,  
 DB 00H - признак конца строки,

где MT0 (MT1) – адрес, заносимый в регистровую пару [HL] перед вызовом подпрограммы 0C818H, MT0 – при начальном вызове и индикации 0000d, MT1 – при обновлении показаний счетчика. Ячейки строки располагаются в произвольном месте ОЗУ пользователя.

### Задание 1.5

Реализовать счетчик событий используя параллельный периферийный адаптер K580BB55A микропроцессорной системы. Регистры ППА имеют следующие адреса в совмещенной карте памяти:

регистр А - 0FFE0H;  
 регистр В - 0FFE1H;  
 регистр С - 0FFE2H;  
 регистр управляющего слова - 0FFE3H.

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ППА: режим 0 – простой ввод/вывод,

канал А – вывод,  
 канал В – ввод,  
 канал С – вывод.

Для подсчета событий использовать разряд 1 канала В ППА, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку датчика соответствует значению "истинно", а установка в единичное состояние – "ложно". Управление счетчиком осуществляется по кодам клавиатуры контроллера, для ввода которых использовать стандартную системную подпрограмму 0C81BH, возвращающую код нажатой клавиши в регистре А микропроцессора без остановки работы программы. Опрос клавиатуры производить во время состояния "ложно". Управляющий код 0DH ("Enter") устанавливает счетчик в состояние 0000d, код 1BH ("Esc") инициализирует выход в системный монитор по адресу 0F800H, код 0FFH свидетельствует о том, что ни одна клавиша не нажата.

Для вывода текущего состояния счетчика на дисплей использовать системную подпрограмму 0C818H, осуществляющую вывод строки символов на экран по следующему соглашению:

MT0:DB 1FH  
 MT1:DB 0CH  
 DB 20H  
 MS3:DB 30H - разряд 3 (старший) счетчика,  
 MS2:DB 30H - разряд 2 счетчика,

**MS1:DB** 30H - разряд 1 счетчика,  
**MS0:DB** 30H - разряд 0 (младший) счетчика,  
**DB** 00H - признак конца строки,

где МТ0 (МТ1) – адрес, заносимый в регистровую пару [HL] перед вызовом подпрограммы 0С818H, МТ0 – при начальном вызове и индикации 0000d, МТ1 – при обновлении показаний счетчика. Ячейки строки располагаются в произвольном месте ОЗУ пользователя.

### Задание 1.6

Реализовать программно визуальный эффект типа "бегущий огонь" со сменой направления движения, используя параллельный периферийный адаптер К580ВВ55А микропроцессорной системы.

К выводам каналов А и В ППА через буферные регистры–формирователи КР580ИР83 подключена линейка из шестнадцати светодиодов.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

**регистр А - 0FFE4H;**  
**регистр В - 0FFE5H;**  
**регистр С - 0FFE6H;**  
**регистр управляющего слова - 0FFE7H.**

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH. При выполнении задания реализовать следующую настройку ППА: режим 0 – простой ввод/вывод,

канал А – вывод,

канал В – вывод,

канал С – ввод.

Для выхода из программы использовать разряд 1 канала В ППА по адресу 0FFE1H, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку (клавиша "Shift") инициализирует безусловный переход в системный монитор по адресу 0F800H.

### Задание 1.7

Реализовать программно линейный светодиодный индикатор с увеличением и уменьшением длины светящейся линии, используя параллельный периферийный адаптер К580ВВ55А микропроцессорной системы.

К выводам каналов А и В ППА через буферный регистр–формирователь КР580ИР83 подключена линейка из шестнадцати светодиодов. Регистры ППА имеют следующие адреса в совмещенной карте памяти:

**регистр А - 0FFE4H;**  
**регистр В - 0FFE5H;**  
**регистр С - 0FFE6H;**  
**регистр управляющего слова - 0FFE7H.**



Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ППА:  
 режим 0 – простой ввод/вывод,  
 канал А – ввод,  
 канал В – вывод,  
 канал С – вывод.

Для выхода из программы использовать опрос разряда 1 канала В ППА по адресу 0FFE1H, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку (клавиша “Shift”) инициализирует безусловный переход в системный монитор по адресу 0F800H.

### Задание 1.8

Осуществить генерацию симметричных пилообразных аналоговых импульсов амплитудой 1 В с помощью параллельного периферийного адаптер K580BV55A микропроцессорной системы.

К выводам канала А ППА через буферный регистр–формирователь КР580ИР82 подключен цифро–аналоговый преобразователь (ЦАП) лестничного типа с матрицей R–2R прецизионных резисторов на входе. Изменение кода на выводах канала А ППА в диапазоне 0÷255 вызывает изменение аналогового сигнала на выходе ЦАП от 0 до 1 В.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

регистр А – 0FFE4H  
 регистр В – 0FFE5H  
 регистр С – 0FFE6H  
 регистр управляющего слова – 0FFE7H.

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ППА:  
 режим 0 - простой ввод/вывод,  
 канал А - вывод,  
 канал В - ввод,  
 канал С - ввод.

Для выхода из программы использовать опрос разряд 1 канала В ППА по адресу 0FFE1H, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку (клавиша “Shift”) инициализирует безусловный переход в системный монитор по адресу 0F800H.

**Задание 1.9**

Осуществить программное управление светодиодным 8–разрядным 7–сегментным индикатором, используя параллельный периферийный адаптер K580BB55A микропроцессорной системы.

К выводам 0÷7 канала А ППА через буферные формирователи подключены соответственно сегменты a, b, c, d, e, f, g индикатора. Логическая единица в соответствующем разряде вызывает свечение сегмента. Подсветка разрядов индикатора осуществляется через выводы 0÷2 канала В, управляющие работой дешифратора типа "код–позиция". Используя принцип динамической индикации вывести порядковый номер наступающего нового года в разрядах 2÷5 индикатора.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

**регистр А - 0FFE4H;**  
**регистр В - 0FFE5H;**  
**регистр С - 0FFE6H;**  
**регистр управляющего слова - 0FFE7H.**

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ППА: режим 0 – простой ввод/вывод,

канал А – вывод,  
 канал В – вывод,  
 канал С – ввод.

Для выхода из программы использовать опрос разряда 1 канала В ППА по адресу 0FFE1H, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку инициализирует безусловный переход в системный монитор по адресу 0F800H.

**Задание 1.10**

Осуществить программное управление светодиодным 8–разрядным 7–сегментным индикатором, используя параллельный периферийный адаптер K580BB55A микропроцессорной системы.

К выводам 0÷7 канала А ППА через буферные формирователи подключены соответственно сегменты a, b, c, d, e, f, g индикатора. Логическая единица в соответствующем разряде вызывает свечение сегмента. Подсветка разрядов индикатора осуществляется через выводы 0÷2 канала В, управляющие работой дешифратора типа "код–позиция". Используя принцип динамической индикации вывести на индикатор сообщение HELLO–XXXX в виде бегущей строки. Число XXXX — порядковый номер наступающего нового года.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

регистр А - 0FFE4H  
 регистр В - 0FFE5H  
 регистр С - 0FFE6H  
 регистр управляющего слова - 0FFE7H.

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ППА:  
 режим 0 – простой ввод/вывод,

канал А – вывод,  
 канал В – вывод,  
 канал С – ввод.

Для выхода из программы использовать опрос разряда 1 канала В ППА по адресу 0FFE1H, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку инициализирует безусловный переход в системный монитор по адресу 0F800H.

### Задание 1.11

Осуществить вывод на печать ассемблерного текста данной программы, используя параллельный периферийный адаптер K580BB55A микропроцессорной системы для управления матричным принтером.

К выводам 0÷7 канала А ППА через буферный формирователь подключены соответственно выводы D0÷D7 интерфейса принтера. Через бит 7 порта В ППА осуществляется контроль за состоянием линии **Busy** принтера, сигнал  $\overline{\text{STROBE}}$  выдается через бит 0 порта С.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

регистр А - 0FFE4H;  
 регистр В - 0FFE5H;  
 регистр С - 0FFE6H;  
 регистр управляющего слова - 0FFE7H.

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

Текстовый буфер редактора расположен в памяти начиная с адреса 1100H. Конец текстового буфера индицирует стоп-байт 0FFH. Печать текста программы осуществить шрифтом Courier bold, кегль 8÷10.

При выполнении задания реализовать следующую настройку ППА:  
 режим 0 – простой ввод/вывод,

канал А – ввод,  
 канал В – вывод,  
 канал С – вывод.

Для выхода из цикла печати использовать опрос разряда 1 канала В ППА по адресу 0FFE1H, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку (клавиша “Shift”) инициализи-

рует выход из подпрограммы печати. Для управления программой печати использовать системную подпрограмму опроса консоли 0С81ВН, возвращающую код нажатой клавиши в регистре А микропроцессора или код 0FFН, в случае если ни одна из клавиш не нажата. Для запуска программы печати использовать код клавиши пробел (20Н), для прекращения процесса печати и выхода в системный монитор по адресу 0F800Н использовать код клавиши “Esc” — 1ВН.

### Задание 1.12

Осуществить печать графической копии экрана дисплея, используя параллельный периферийный адаптер К580ВВ55А микропроцессорной системы для управления матричным принтером.

К выводам 0÷7 канала А ППА через буферный формирователь подключены соответственно выводы D0÷D7 интерфейса принтера. Через бит 7 порта В ППА осуществляется контроль за состоянием линии **Busy** принтера, сигнал **STROBE** выдается через бит 0 порта С.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

регистр А - 0FFE4Н;  
 регистр В - 0FFE5Н;  
 регистр С - 0FFE6Н;  
 регистр управляющего слова - 0FFE7Н.

Доступное пользователю ОЗУ расположено по адресам 0000Н÷7FFFН, адреса видео-ОЗУ: 9000Н÷0BFFFН. Видео-ОЗУ имеет следующую организацию: 48 байтовых столбцов по 256 строк. Адрес левого верхнего байта ОЗУ экрана – 9000Н, адрес левого нижнего байта ОЗУ экрана – 90FFН. Адрес следующего байта в верхней строке – 9100Н и т.д. Экранное ОЗУ заканчивается 48-м столбцом, в котором верхний байт расположен по адресу 0BF00Н, а нижний – по адресу 0BFFFН.

При выполнении задания реализовать следующую настройку ППА: режим 0 – простой ввод/вывод,

канал А – ввод,  
 канал В – вывод,  
 канал С – вывод.

Для выхода из цикла печати использовать опрос разряда 1 канала В ППА по адресу 0FFE1Н, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку инициализирует выход из подпрограммы печати. Для управления программой печати использовать системную подпрограмму опроса консоли 0С81ВН, возвращающую код нажатой клавиши в регистре А микропроцессора или код 0FFН, в случае если ни одна из клавиш не нажата. Для запуска программы печати использовать код клавиши пробел (20Н), для прекращения процесса печати и вы-

хода в системный монитор по адресу 0F800H использовать код клавиши “Esc” — 1BH.

Информацию об установке текущего шрифта печатающего устройства взять из руководства по эксплуатации подключенного к системе принтера.

### Задание 1.13

Осуществить ввод информации с цифрового выхода аналого–цифрового преобразователя (АЦП) типа Ф–214 (Ф–295) в параллельном коде 8–4–2–1, используя параллельный периферийный адаптер K580BV55A микропроцессорной системы, и вывести значение на дисплей.

К выводам 0÷3 канала А ППА подключены выходы разряда  $10^0$  АЦП, к выводам 4÷7 канала А ППА подключены выходы разряда  $10^1$  АЦП, к выводам 0÷3 канала В ППА подключены выходы разряда  $10^2$  АЦП, к выводам 4÷7 канала В ППА подключены соответственно выходы разряда  $10^3$  АЦП. Управление АЦП осуществляется через бит 7 канала С. Установка уровня “1” по этому биту и его удержание в течение более 2 мкс разрешают АЦП провести одиночный цикл преобразования. Установка “1” в разряде 0 порта С ППА свидетельствует о начале цикла измерения. Уровнем “0” через бит 0 порта С ППА АЦП подтверждает завершение измерительного цикла.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

регистр А - 0FFE4H  
 регистр В - 0FFE5H  
 регистр С - 0FFE6H  
 регистр управляющего слова - 0FFE7H.

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ППА: режим 0 – простой ввод/вывод,

канал А – ввод,  
 канал В – ввод,  
 канал С0÷С3 – ввод,  
 канал С4÷С7 – вывод.

Для вывода текущего состояния счетчика на дисплей использовать системную подпрограмму 0C818H, осуществляющую вывод строки символов на экран по следующему соглашению:

|        |     |                                |
|--------|-----|--------------------------------|
| MT0:DB | 1FH |                                |
| MT1:DB | 0CH |                                |
| DB     | 20H |                                |
| MS3:DB | 30H | - разряд $10^3$ (старший) АЦП, |
| MS2:DB | 30H | - разряд $10^2$ АЦП,           |
| MS1:DB | 30H | - разряд $10^1$ АЦП,           |
| MS0:DB | 30H | - разряд $10^0$ (младший) АЦП, |
| DB     | 00H | - признак конца строки,        |

где МТ0 (МТ1) – адрес, заносимый в регистровую пару [HL] перед вызовом подпрограммы 0С818H, МТ0 – при начальном вызове и индикации 0000d, МТ1 – при обновлении показаний АЦП. Ячейки строки располагаются в произвольном месте ОЗУ пользователя.

Для выхода из программы использовать опрос разряда 1 канала В ППА по адресу 0FFE1H, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку инициализирует безусловный переход в системный монитор по адресу 0F800H.

### **Задание 1.14**

Осуществить ввод аналоговой информации в ОЗУ микропроцессорной системы методом единичного приближения с последующим выводом через параллельный периферийный адаптер K580BB55A.

К выводам канала А ППА через буферный регистр–формирователь КР580ИР82 подключен цифро–аналоговый преобразователь лестничного типа с матрицей R–2R прецизионных резисторов на входе. Изменение кода на выводах канала А ППА в диапазоне 0÷255 вызывает изменение аналогового сигнала на выходе ЦАП от 0 до 1 В.

Аналоговый компаратор, выход которого подключен к разряду 2 канала В ППА, осуществляет сравнение входного аналогового сигнала частотой до 3,5 кГц и амплитудой в диапазоне 0÷1 В с опорным сигналом ЦАП. Превышение опорным сигналом величины входного сигнала приводит к появлению логического уровня "1" на выходе компаратора.

Регистры ППА имеют следующие адреса в совмещенной карте памяти:

**регистр А - 0FFE4H**

**регистр В - 0FFE5H**

**регистр С - 0FFE6H**

**регистр управляющего слова - 0FFE7H.**

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ППА:

режим 0 - простой ввод/вывод,

канал А - вывод,

канал В - ввод,

канал С - ввод.

Для управления программой использовать опрос разряда 1 канала В ППА, установка которого в нулевое состояние через подключенную к нему замыкающую кнопку (клавиша "Shift") инициализирует процессы ввода и последующего вывода аналогового сигнала.

**Примечание.** При выполнении заданий с использованием подпрограмм-имитаторов внешних устройств, а также в отдельных случаях специальной настройки конфигурации программы эмулятора адреса регистров ППА могут отличаться от приведенных в тексте задания.

## 2.3. ПОРЯДОК ВЫПОЛНЕНИЯ ЗАДАНИЙ

1. Подготовить компьютер для выполнения соответствующего задания.
2. Ввести в редакторе подготовленные тексты программ, скомпилировать объектный код, отладить и выполнить программу. При статической отладке, следует в пошаговом режиме убедиться в правильности загрузки управляющих слов и рабочих значений в соответствующие регистры ППА, проконтролировать вывод в порты многоканальным логическим пробником. В режиме динамической отладки форму сигналов и правильность выполнения алгоритма следует проверить при помощи осциллографа или многоканального логического анализатора.
3. Оформить индивидуальный отчет, включающий необходимые иллюстративные материалы (функциональные схемы исследуемых подсистем, алгоритмы разработанных программ, временные диаграммы) и листинги отлаженных программ.

## СПИСОК ЛИТЕРАТУРЫ

1. Гук М. Аппаратные средства IBM-PC. Энциклопедия, 2-е изд. – СПб.: Питер 2003. – 928 с.
2. Микропроцессоры и микропроцессорные комплекты интегральных микросхем. Под ред. В.А.Шахнова. – М.: "Радио и связь", 1988. Т.1, – 368 с.
3. Микропроцессоры. В 3-х кн. / П.В.Нестеров, В.Ф.Шаньгин, В.Л.Горбунов и др.; Под ред. Л.Н.Преснухина. – М.: "Высшая школа", 1986.
4. Гутников В.С. Интегральная электроника в измерительных устройствах. – Ленинград.: "Энергоатомиздат", 1988. – 304 с.
5. Васильев Б.И., Гусев Ю.М., Миронов В.Н. Электронные промышленные устройства. – М.: "Высшая школа", 1988. – 304 с.
6. Сопряжение датчиков и устройств ввода данных с компьютерами IBM PC: Пер. с англ. / Под ред. У. Томпкинса и Дж. Уэбстера. – М.: Мир, 1992. – 592 с.
7. Нортон П., Соухэ Д. Язык ассемблера для IBM PC. – М.: "Компьютер", 1993. – 352 с.
8. Рудаков П.И., Финогенов К.Г. Программируем на языке ассемблера IBM PC: В 4-х частях. Ч.1. (Ч.2., Ч.3–4.) – М.: "Энтроп", 1995. – 164 с. (164 с., 320 с.).

## ПРИЛОЖЕНИЕ

СИСТЕМА КОМАНД МИКРОПРОЦЕССОРА K580BM80

Флаги МП образуют регистр флагов (признаков) F: S.Z.0.AC.0.P.1.C

Условные обозначения: Z (ZERO) - флаг нулевого результата

S (SIGN) - флаг знака результата,

C (CARRY) - флаг переноса,

AC (AXIAL CARRY) - флаг осевого переноса,

P (PARITY) - флаг четности результата,

+ - команда изменяет флаг,

- - команда не изменяет флаг,

0 - команда устанавливает флаг в 0,

1 - команда устанавливает флаг в 1,

R - регистр общего назначения,

RP - регистровая пара,

# - однобайтовый операнд,

## - двухбайтовый операнд,

XX - адрес.

| КОМАНДА | ДЛИНА<br>(БАЙТ) | ФУНКЦИЯ  | ПРИЗНАКИ |   |   |    |   |
|---------|-----------------|--|----------|---|---|----|---|
| 1       | 2               | 3  | Z        | S | C | AC | P |
| MOV R,R | 1               | передача содержимого одного регистра в другой                                    | -        | - | - | -  | - |
| MVI R,# | 2               | загрузка регистра вторым байтом команды  | -        | - | - | -  | - |
| INR R   | 1               | увеличить содержимое регистра на 1   | +        | + | - | +  | + |
| DCR R   | 1               | уменьшить содержимое регистра на 1   | +        | + | - | +  | + |
| ADD R   | 1               | к содержимому аккумулятора прибавить содержимое регистра                         | +        | + | + | +  | + |
| ADC R   | 1               | к содержимому аккумулятора прибавить содержимое регистра с учетом флага переноса | +        | + | + | +  | + |
| SUB R   | 1               | вычесть из аккумулятора содержимое регистра                                      | +        | + | + | +  | + |
| SBB R   | 1               | вычесть из аккумулятора содержимое регистра и флаг переноса                      | +        | + | + | +  | + |
| ANA R   | 1               | логическое умножение аккумулятора и регистра                                     | +        | + | 0 | +  | + |
| XRA R   | 1               | выполнить операцию неравнозначности аккумулятора и регистра                      | +        | + | 0 | 0  | + |
| ORA R   | 1               | выполнить операцию логического сложения аккумулятора и регистра                  | +        | + | 0 | 0  | + |
| CMP R   | 1               | сравнить содержимое аккумулятора и регистра                                      | +        | + | + | +  | + |
| ADI #   | 2               | к содержимому аккумулятора прибавить второй байт команды                         | +        | + | + | +  | + |
| ACI #   | 2               | к содержимому аккумулятора прибавить второй байт команды с учетом флага переноса | +        | + | + | +  | + |



| 1       | 2 | 3   | Z | S | C | AC | P |
|---------|---|---|---|---|---|----|---|
| SUI #   | 2 | из содержимого аккумулятора вычесть второй байт команды                         | + | + | + | +  | + |
| SBI #   | 2 | из содержимого аккумулятора вычесть второй байт команды с учетом флага переноса | + | + | + | +  | + |
| ANI #   | 2 | выполнить операцию логического умножения аккумулятора и второго байта команды   | + | + | 0 | 0  | + |
| XRI #   | 2 | выполнить операцию неравнозначности аккумулятора и второго байта команды        | + | + | 0 | 0  | + |
| ORI #   | 2 | выполнить логическое сложение аккумулятора и второго байта команды              | + | + | 0 | 0  | + |
| CPI #   | 2 | сравнить содержимое аккумулятора и второго байта команды                        | + | + | + | +  | + |
| RLC     | 1 | сдвинуть циклически влево содержимое аккумулятора                               | - | - | + | -  | - |
| RRC     | 1 | сдвинуть циклически вправо содержимое аккумулятора                              | - | - | + | -  | - |
| RAL     | 1 | сдвинуть циклически влево содержимое аккумулятора через флаг переноса           | - | - | + | -  | - |
| RAR     | 1 | сдвинуть циклически вправо содержимое аккумулятора через флаг переноса          | - | - | + | -  | - |
| JMP XX  | 3 | безусловный переход по указанному адресу  | - | - | - | -  | - |
| JC XX   | 3 | условный переход при единичном состоянии флага переноса                         | - | - | - | -  | - |
| JNC XX  | 3 | условный переход при нулевом состоянии флага переноса                           | - | - | - | -  | - |
| JZ XX   | 3 | условный переход по нулевому значению результата                                | - | - | - | -  | - |
| JNZ XX  | 3 | условный переход по ненулевому значению результата                              | - | - | - | -  | - |
| JP XX   | 3 | условный переход по положительному значению результата                          | - | - | - | -  | - |
| JM XX   | 3 | условный переход по отрицательному значению результата                          | - | - | - | -  | - |
| JPE XX  | 3 | условный переход по четности кода результата                                    | - | - | - | -  | - |
| JPO XX  | 3 | условный переход по нечетности кода результата                                  | - | - | - | -  | - |
| CALL XX | 3 | безусловный вызов подпрограммы по адресу XX                                     | - | - | - | -  | - |
| CC XX   | 3 | вызов подпрограммы при единичном состоянии флага переноса                       | - | - | - | -  | - |
| CP XX   | 3 | вызов подпрограммы при положительном значении результата                        | - | - | - | -  | - |
| CZ XX   | 3 | вызов подпрограммы при нулевом значении результата                              | - | - | - | -  | - |

| 1          | 2 | 3  | Z | S | C | AC | P |
|------------|---|--|---|---|---|----|---|
| CNZ XX     | 3 | вызов подпрограммы при ненулевом значении результата           | - | - | - | -  | - |
| CM XX      | 3 | вызов подпрограммы при отрицательном значении результата       | - | - | - | -  | - |
| CPE XX     | 3 | вызов подпрограммы по четности кода результата                 | - | - | - | -  | - |
| CPO XX     | 3 | вызов подпрограммы по нечетности кода результата               | - | - | - | -  | - |
| RET        | 1 | безусловный возврат из подпрограммы                            | - | - | - | -  | - |
| RC         | 1 | возврат из подпрограммы по единичному состоянию флага переноса | - | - | - | -  | - |
| RP         | 1 | возврат из подпрограммы по положительному результату           | - | - | - | -  | - |
| RM         | 1 | возврат из подпрограммы по отрицательному результату           | - | - | - | -  | - |
| RPE        | 1 | возврат из подпрограммы по четности кода результата            | - | - | - | -  | - |
| RNC        | 1 | возврат из подпрограммы по ненулевому состоянию флага переноса | - | - | - | -  | - |
| RZ         | 1 | возврат из подпрограммы по нулевому результату                 | - | - | - | -  | - |
| RNZ        | 1 | возврат из подпрограммы по ненулевому результату               | - | - | - | -  | - |
| RPO        | 1 | возврат из подпрограммы по нечетности кода результата          | - | - | - | -  | - |
| RST N      | 1 | безусловный вызов подпрограммы по фиксированному адресу (N*8)H | - | - | - | -  | - |
| LXI RP, ## | 3 | загрузка регистровой пары вторым и третьим байтами команды     | - | - | - | -  | - |
| PUSH RP    | 1 | запись регистровой пары в стек                                 | - | - | - | -  | - |
| POP RP     | 1 | загрузка регистровой пары из стека                             | - | - | - | -  | - |
| POP PSW    | 1 | загрузка аккумулятора и регистра признаков из стека            | + | + | + | +  | + |
| STA XX     | 3 | запись содержимого аккумулятора по указанному адресу           | - | - | - | -  | - |
| LDA XX     | 3 | загрузка аккумулятора содержимым ячейки с указанным адресом    | - | - | - | -  | - |
| XCHG       | 1 | поменять местами содержимое регистровых пар [HL] и [DE]        | - | - | - | -  | - |
| XTHL       | 1 | поменять местами содержимое верхней ячейки стека и [HL]        | - | - | - | -  | - |
| SPHL       | 1 | загрузить в указатель стека [SP] содержимым [HL]               | - | - | - | -  | - |
| PCHL       | 1 | загрузить программный счетчик [PC] содержимым [HL]             | - | - | - | -  | - |

| 1       | 2 | 3  | Z | S | C | AC | P |
|---------|---|--|---|---|---|----|---|
| DAD RP  | 1 | к содержимому [HL] прибавить содержимое другой регистровой пары или указателя стека и результат поместить в [HL] | - | - | + | -  | - |
| STAX RP | 1 | записать содержимое аккумулятора по адресу, указанному в регистровой паре  | - | - | - | -  | - |
| LDAX RP | 1 | загрузить аккумулятор содержимым ячейки, адрес которой указан в регистровой паре                                 | - | - | - | -  | - |
| INX RP  | 1 | увеличить на 1 содержимое регистровой пары или указателя стека   | - | - | - | -  | - |
| DCX RP  | 1 | уменьшить на 1 содержимое регистровой пары или указателя стека   | - | - | - | -  | - |
| CMA     | 1 | инвертировать содержимое аккумулятора  | - | - | - | -  | - |
| STC     | 1 | установить флаг переноса в 1   | - | - | 1 | -  | - |
| CMC     | 1 | инвертировать флаг переноса  | - | - | + | -  | - |
| DAA     | 1 | десятичная коррекция результата  | + | + | + | +  | + |
| SHLD XX | 3 | записать содержимое [HL] в две ячейки последовательно с указанного адреса: в первую - L, во вторую - H           | - | - | - | -  | - |
| LHLD XX | 3 | загрузить пару [HL] содержимым двух последовательных ячеек с указанного адреса, из младшей - L, из старшей - H   | - | - | - | -  | - |
| NOP     | 1 | пустая операция  | - | - | - | -  | - |
| HLT     | 1 | останов процессора   | - | - | - | -  | - |
| IN #    | 2 | загрузка аккумулятора содержимым порта с номером #   | - | - | - | -  | - |
| OUT #   | 2 | запись содержимого аккумулятора в порт с номером #   | - | - | - | -  | - |
| DI      | 1 | запретить прерывания   | - | - | - | -  | - |
| EI      | 1 | разрешить прерывания   | - | - | - | -  | - |



Учебное издание

*Семёнов Андрей Андреевич*

ИЗУЧЕНИЕ БИС  
ПАРАЛЛЕЛЬНОГО  
ПЕРИФЕРИЙНОГО  
АДАПТЕРА

Учебное пособие  
для студентов факультета компьютерных наук  
и информационных технологий  
и факультета nano- и биомедицинских технологий

Редактор В.А. Т р у ш и н а  
Технический редактор Л.В. А г а л ь ц о в а  
Корректор Ю.И. А с т а х о в а

---

Подписано в печать \_\_.\_\_.2006.  
Формат 60×84 1/16. Бумага офсетная. Гарнитура Times. Печать офсетная.  
Усл.печ.л.1,16(1,25). Уч.-изд.л.0,9. Тираж 150 экз. Заказ

---

Издательство Саратовского университета.  
410012, Саратов, Астраханская, 83.  
Типография Издательства Саратовского университета.  
410012, Саратов, Астраханская, 83.