

МИНИСТЕРСТВО АТОМНОЙ ЭНЕРГИИ

**САРОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
МОСКОВСКОГО ГОСУДАРСТВЕННОГО
ИНЖЕНЕРНО-ФИЗИЧЕСКОГО ИНСТИТУТА
(ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА)**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ЭЛЕКТРОНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ И ИНФОРМАЦИОННОЙ ТЕХНИКИ

**АДАПТЕРЫ И КОНТРОЛЛЕРЫ ЭВМ.
СБОРНИК ЛАБОРАТОРНЫХ РАБОТ**

Рекомендовано УМО вузов
по университетскому техническому образованию
в качестве учебного пособия
для студентов высших учебных заведений
обучающихся по специальности 230101
«Вычислительные машины, комплексы системы и сети»

**Саров
1999**

УДК 681.34
П12

Павлов В. А.

Адаптеры и контроллеры ЭВМ.
Сборник лабораторных работ по курсу

Приводятся описания семи лабораторных работ, посвященных изучению принципов функционирования и программирования отдельных компонентов систем ввода-вывода персональных ЭВМ.

Сборник предназначен для студентов старших курсов технических вузов.

Утверждено:
Научно-методический совет
СарФТИ
30. 09. 1999г.

Содержание

ВВЕДЕНИЕ	6
1.АДАПТЕР ААПД.	7
1.1. ВВЕДЕНИЕ.....	8
1.2. НАЗНАЧЕНИЕ ААПД	8
1.3. ТЕХНИЧЕСКИЕ ДАННЫЕ	8
1.4. СОСТАВ ААПД.....	12
1.5. УСТРОЙСТВО И РАБОТА ААПД.....	12
1.6. УСТРОЙСТВО И РАБОТА СОСТАВНЫХ ЧАСТЕЙ ААПД	16
1.7. КОНСТРУКЦИЯ ААПД.....	20
1.8. МАРКИРОВКА.....	21
1.9. ЭКСПЛУАТАЦИЯ ААПД	21
1.10. УКАЗАНИЕ МЕР БЕЗОПАСНОСТИ	21
1.11. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ	21
1.12. ПРИЛОЖЕНИЯ	21
2. <u>ЛАБОРАТОРНАЯ РАБОТА 1. «БИС ПРОГРАММИРУЕМОГО ТАЙМЕРА</u>	
КР1810ВИ54(І8254)».....	28
2.1. ЦЕЛЬ РАБОТЫ :	29
2.2. ТЕХНИЧЕСКИЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ	29
В ЛАБОРАТОРНОЙ РАБОТЕ :	29
2.3. НАЗНАЧЕНИЕ ПРОГРАММИРУЕМОГО ТАЙМЕРА КР1810ВИ54 (І 8254) :	29
2.4. ФУНКЦИОНИРОВАНИЕ КР1810ВИ54 (І 8254) :	29
2.5. ПОДГОТОВКА К ЛАБОРАТОРНОЙ РАБОТЕ	31
2.6. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ :	31
2.7. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА :	31
2. 8. ПРИЛОЖЕНИЯ :	32
2.8.1. ОПИСАНИЕ КР1810ВИ54 (і 8254) :	32
2.8.2. Примеры программ	39
2.8.3. Примеры временных диаграмм работы программируемого таймера в разных режимах	45
3. <u>ЛАБОРАТОРНАЯ РАБОТА 2. «УНИВЕРСАЛЬНЫЙ СИНХРОННО-</u>	
АСИНХРОННЫЙ ПРИЕМОПЕРЕДАТЧИК (УСАПП) КР580ВВ51 В СОСТАВЕ	
АДАПТЕРА АПД ПЭВМ» ИСКРА 1031".....	51
3.1. ЦЕЛЬ РАБОТЫ.....	52
3.2. ТЕХНИЧЕСКИЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ В РАБОТЕ.....	52
3.3. ОСНОВНЫЕ ЭЛЕКТРИЧЕСКИЕ ПАРАМЕТРЫ БИС КР580ВВ51.....	52
3.4. НАЗНАЧЕНИЕ УСАПП КР580ВВ51.	52
3.5. УКАЗАНИЕ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ.....	55
3.6. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ	55
3.7. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.	55
3.8.ПРИЛОЖЕНИЕ	56
3.8.1 Структурная схема УСААП.....	56
3.8.2 Функционирование КР580ВВ51.....	58

3.8.3. Команды	60
3.8.4. Формат слова-состояния.....	61
3.8.5. Режимы работы.	61
3.8.6. Пример программы.	61
3.8.7. Осциллограммы на выходе УСАПП	63
4.ЛАБОРАТОРНАЯ РАБОТА 3.«БИС ПАРАЛЛЕЛЬНОГО ПРОГРАММИРУЕМОГО ИНТЕРФЕЙСА КР580ВВ55 В СОСТАВЕ ААПД»	66
4.1 . ЦЕЛЬ РАБОТЫ.....	67
4.2 . ТЕХНИЧЕСКИЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ В РАБОТЕ.	67
4.3 . ОСНОВНЫЕ ЭЛЕКТРИЧЕСКИЕ ПАРАМЕТРЫ БИС КР580ВВ55.....	67
4.4 . НАЗНАЧЕНИЕ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРУЕМОГО ИНТЕРФЕЙСА КР580ВВ55.....	67
4.5 . ФУНКЦИОНИРОВАНИЕ ППИ К580ВВ55.	67
4.6 . УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ.....	68
4.7. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ.	69
4.8. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.	70
4.9. ПРИЛОЖЕНИЯ.	70
4.9.1. Описание ППИ КР580ВВ55:	70
4.9.2 Примеры программ	77
4.9.3. Распределение сигналов по контактам разъема коннектора	78
5.ЛАБОРАТОРНАЯ РАБОТА 4. «ПАРАЛЛЕЛЬНЫЙ ПОРТ IBM PC/AT».....	79
5.1. ЦЕЛЬ РАБОТЫ.....	80
5.2. ТЕХНИЧЕСКИЕ СРЕДСТВА , ИСПОЛЬЗУЕМЫЕ В РАБОТЕ	80
5.3. ПРОГРАММИРОВАНИЕ ПАРАЛЛЕЛЬНОГО ПОРТА	80
5.4. ПРЕРЫВАНИЯ ПРИНТЕРА.....	80
5. 5. ПРЕРЫВАНИЯ ПРИНТЕРА ЧЕРЕЗ ФУНКЦИИ BIOS-A.	81
5.6. ДРАЙВЕР ПРИНТЕРА INT 17H.....	81
5.7. УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ.....	82
5.8. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ	82
5.9. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	83
5.10. ПРИЛОЖЕНИЯ	83
5.10.1 Описание контроллера параллельного порта.....	83
5.10.2 Структурная схема двунаправленного параллельного порта	83
5.10.3 Описание регистров параллельного порта.	83
5.10.4 Примеры программ.	86
6.ЛАБОРАТОРНАЯ РАБОТА 5. « ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ IBM PC/AT».....	88
6.1. ЦЕЛЬ РАБОТЫ.....	89
6.2.ТЕХНИЧЕСКИЕ СРЕДСТВА.	89
6.3. НАЗНАЧЕНИЕ КОНТРОЛЛЕРА ПОСЛЕДОВАТЕЛЬНОГО ИНТЕРФЕЙСА.....	89
6.4. ПРИНЦИПЫ РАБОТЫ RS-232C.	89
6.5. ПОДГОТОВКА К ЛАБОРАТОРНОЙ РАБОТЕ.....	90
6.6. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ	90
6.7. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.	91
6.8. ПРИЛОЖЕНИЕ	91
6.8.1 Сигналы интерфейса RS-232C.....	91
6.8.2 Назначение сигналов интерфейса RS-232C	92
6.8.3 Подготовка интерфейса RS-232C к работе	92

6.8.4 Принципы построения универсального асинхронного приемопередатчика (УАПП).....	93
6.8.5 Инициализация УАПП.....	94
6.8.6 Структура регистров УАПП.....	95
6.8.7 Прерывания RS232.....	103
6.8.8 Основы программирования контроллера последовательного интерфейса	107
7. ЛАБОРАТОРНАЯ РАБОТА 6. «ПРОГРАММИРОВАНИЕ КЛАВИАТУРЫ»	112
7.1. ЦЕЛЬ РАБОТЫ.....	113
7.2. ТЕХНИЧЕСКИЕ СРЕДСТВА , ИСПОЛЬЗУЕМЫЕ В РАБОТЕ	113
7.3. ОПИСАНИЕ.....	113
7.3.1 Клавиатура.....	113
7.3.2 Функционирование клавиатуры.	113
7.3.3 Программирование клавиатуры.....	115
7.3.4 Режимы работы клавиатуры.	118
7.3.5 Регистры и порты контроллера клавиатуры.	119
7.3.6 Команды контроллера клавиатуры.....	121
7.4. ПОДГОТОВКА К ЛАБОРАТОРНОЙ РАБОТЕ	124
7.5. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ	124
7.6. СПИСОК ЛИТЕРАТУРЫ.....	124
7.7. ПРИЛОЖЕНИЯ.	124
7.7.1. Примеры программ	124
8.ЛАБОРАТОРНАЯ РАБОТА 7. «ПРОГРАММИРОВАНИЕ RTC CMOS»	127
8.1 . ЦЕЛЬ РАБОТЫ.....	128
8.2 . ТЕХНИЧЕСКИЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ В РАБОТЕ.	128
8.3 . НАЗНАЧЕНИЕ МИКРОСХЕМЫ MC146818(RTC/CMOS).....	128
8.4 . ФУНКЦИОНИРОВАНИЕ МИКРОСХЕМЫ MC146818(RTC/CMOS).	128
8.5 . ПРОГРАММИРОВАНИЕ МИКРОСХЕМЫ MC146818(RTC/CMOS).....	130
8.6 . УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ.....	130
8.7 . ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ.	131
8.8 . РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.	131
8.9 . ПРИЛОЖЕНИЯ.	132
8.9.1.Описание микросхемы MC146818(RTC/CMOS).	132
8.9.2.Содержимое обработчика прерываний программы BEEP_TSR.EXE.	143
8.9.3. Примеры программ	144
9. ПРИЛОЖЕНИЕ	147
9. 1. ОСНОВНЫЕ КОМАНДЫ ОТЛАДЧИКА AFD	148
9.2 . СИСТЕМА КОМАНД МИКРОПРОЦЕССОРА K1810 VM86 (I8086)	150
9.3. СПИСОК ЛИТЕРАТУРЫ, ИСПОЛЬЗУЕМОЙ ПРИ ПОДГОТОВКЕ К ЛАБОРАТОРНЫМ РАБОТАМ	173

ВВЕДЕНИЕ

Важное значение в подготовке специалистов в области вычислительной и информационной техники является изучение принципов работы и программирования системы ввода-вывода (СВВ) персональных компьютеров на самом нижнем - физическом уровне.

Настоящий сборник лабораторных работ по курсу «Адаптеры и контроллеры ЭВМ» содержит описание отдельных компонентов СВВ персональных компьютеров, принципов их работы и программирования.

В процессе выполнения работ студентами предусматривается написание на ассемблере и отладка в среде AFD (полноэкранный отладчик) простых программ инициализации подсистем СВВ и управление обменом через порты ввода-вывода.

За работой программ и подсистем предусматривается наблюдать с помощью измерительных приборов (осциллографы, вольтметры и пр.), а также косвенным путем (индикаторы, вспомогательные программы и пр.).

В сборнике умышленно допущены ошибки в примерах программ, реализующих те или иные функции инициализации подсистем СВВ и управления вводом-выводом информации. Т.е. каждая приведенная программа требует отладки, которую можно успешно провести только изучив принципы работы и программирования соответствующих подсистем СВВ.

В первой главе приведено описание Адаптера Аппаратуры передачи данных ПЭВМ «Искра-1031» (ААПД 3.093.142), выполненного на базе больших интегральных схем (БИС) последовательного (КР580ВВ51) и параллельного (КР580ВВ55) интерфейсов.

Вторая глава посвящена БИС трехканального программируемого таймера КР1810 ВИ54 в составе подсистем таймера ПЭВМ «Искра-1031» и IBM PC/AT.

Третья и шестая главы содержат соответственно описания принципов работы БИС последовательного интерфейса КР580ВВ51 в составе ААПД и последовательного порта IBM PC/AT (СОМ-порт стандарта RS-232C).

Четвертая и пятая главы посвящены соответственно БИС параллельного интерфейса КР580ВВ55 в составе ААПД и параллельного порта IBM PC/AT стандарта Centronics.

В седьмой главе излагаются принципы работы и программирования адаптера клавиатуры и клавиатуры IBM PC/AT.

В восьмой главе изложен материал, посвященный подсистеме часов реального времени RTC/CMOS IBM PC/AT.

В приложении приведены основные команды полноэкранного отладчика AFD а также описание системы команд микропроцессора КР1810 ВМ86 (i8086).

В процессе создания сборника лабораторных работ использовалась программа для настольных издательских систем Page Maker 6.5 и пакет MS Word-97.

При написании разработки были использованы данные из курсовых работ студентов СарФТИ, выполненных в рамках учебного процесса по курсу «Адаптеры и контроллеры ЭВМ» а также дипломной работы Федоркиной А.Н. «Отладка и тестирование лабораторных работ по курсу «Адаптеры и контроллеры ЭВМ», СарФТИ, 1999г.

1 . АДАПТЕР ААПД .

Техническое описание 3.093.138 ТО.

1.1. ВВЕДЕНИЕ

1.1.1 Настоящее техническое описание 3.093.138 ТО (в дальнейшем ТО) предназначено для изучения технических характеристик адаптера АПД 3.093.138 (в дальнейшем ААПД).

1.1.2 При изучении ААПД необходимо пользоваться следующими документами:

1) ГОСТ 18145-81 “Цепи на стыке С2 аппаратуры передачи данных с оконечным оборудованием при последовательном вводе-выводе данных. Номенклатура и технические требования”.

2) НМ МПК по ВТ 10-78 “Система малых электронных вычислительных машин. Интерфейс для радиального подключения устройств с последовательной передачей информации - ИРПС”.

3) Паспорта, технические описания, инструкции по эксплуатации на используемые в ААПД интегральные микросхемы.

4) схемой электрической принципиальной 3.093.183 ЭЗ.

5) сборочным чертежом 3.093.142 СБ.

1.2. НАЗНАЧЕНИЕ ААПД

1.2.1. ААПД предназначен для использования в составе персональных профессиональных электронных вычислительных машин (ПП ЭВМ “Искра 1030.11”).

1.2.2. ААПД предназначен для организации дистанционной передачи данных при:

1) температуре окружающего воздуха от 10 до 50оС;

2) относительной влажности воздуха от 40 до 80% при температуре 25оС;

3) атмосферном давлении от 84 до 107 кПа.

1.3. ТЕХНИЧЕСКИЕ ДАННЫЕ

1.3.1. ААПД в составе технических средств, в которые он входит, обеспечивает возможность одновременной и независимой работы:

1) по шести каналам ИРПС: ИРПС1 - ИРПС6;

2) по телефонному каналу.

1.3.2. ААПД обеспечивает возможность автоматического установления входящего и исходящего соединения по коммутируемым телефонным каналам с помощью устройств автоматического вызова (УАВ), подключаемых к ААПД по стыку С2, серия 200.

1.3.3. ААПД обеспечивает передачу данных по коммутируемым и некоммутируемым телефонным каналам связи с помощью УПС, подключаемых к ААПД по стыку С2, серия 100.

1.3.4. Номенклатура сигналов обмена с УПС и УАВ и размещение их по контактам разъема ААПД приведены соответственно в табл.1 и табл.2.

Уровни сигналов обмена по стыку С2 следующие:

1) состоянию двоичной “1”, “Выкл” соответствует уровень сигнала от -3 до -12 В;

2) состоянию двоичного “0”, “Вкл” соответствует уровень сигнала от 3 до 12 В.

1.3.5. ААПД обеспечивает взаимодействие с удаленными устройствами ввода - вывода по рангу ИРПС. Номенклатура цепей взаимосвязи и их размещение по контактам разъема приведены в табл.3. Состоянию двоичной “1” соответствует ток от 15 до 25 мА, состоянию двоичного “0” со ответствует ток от 0 до 3 мА.

1.3.6. ААПД обеспечивает подключение к системному.

Таблица 1.

<i>Наименование цети</i>	<i>Номер разъема и контакта</i>	<i>Наименование цети</i>	<i>Номер разъема и контакта</i>
101	X3:C20	201	X3:A32
102	X3:A19	202	X3:A20
103	X3:C23	203	X3:C30
104	X3:A28	204	X3:A29
105	X3:C24	205	X3:C29
106	X3:A27	206	X3:C18
107	X3:A26	207	X3:C12
108	X3:C26	208	X3:C15
109	X3:C25	209	X3:C3
114	X3:C27	210	X3:C32
115	X3:A31	211	X3:C9
120	X3:A23	212	X3:C26
125	X3:C31	213	X3:C28
126	X3:C21		

Таблица 2.**Таблица 3**

<i>Наименование цети</i>	<i>Номер разъема и контакта</i>	<i>Наименование цети</i>	<i>Номер разъема и контакта</i>
ПД1+	X3:A1	A0	X1:C24
ПД1-	X3:A2	A1	X1:A24
ПрД1+	X3:C1	A2	X1:C23
ПрД1-	X3:C2	A3	X1:A23
ПД2+	X3:A4	A4	X1:C22
ПД2-	X3:A5	A5	X1:A22
ПрД2+	X3:C4	A6	X1:C21
ПрД2-	X3:C5	A7	X1:A21
ПД3+	X3:A7	A8	X1:C20
ПД3-	X3:A8	A9	X1:A20
ПрД3+	X3:C7	INH3	X1:A7
ПрД3-	X3:C8	Общий провод	X1:A2
ПД4+	X3:A10	IRQ4	X1:C25
ПД4-	X3:A11	IOR $\overline{D}$	X1:C5
ПрД4+	X3:C10	IOWT	X1:A5
ПрД4-	X3:C11	INT	X1:A8
ПД5+	X3:A13	INTA	X1:C12
ПД5-	X3:A14	XACK	X1:C6
ПрД5+	X3:C13	D0	X1:C32
ПрД5-	X3:C14	D1	X1:A32
ПД6+	X3:A16	D2	X1:C31
ПД6-	X3:A17	D3	X1:A31
ПрД6+	X3:C16	D4	X1:C30
ПрД6-	X3:C19	D5	X1:A30
		D6	X1:C29
		D7	

Таблица 4

интерфейсу. Номенклатура логических сигналов обмена с системным интерфейсом и их размещение по контактам разъема Х1 и Х2 приведены в табл.4.

1.3.7. Параметры сигналов обмена ААПД с системным и локальным интерфейсом соответствуют параметрам ИС серий К155, КР580: двоичной "1" соответствует напряжение от 0 до +0,45 В, двоичному "0" соответствует напряжение от 2,4 до 4,8 В.

1.3.8. ААПД обеспечивает байт - ориентированную передачу данных в соответствии с протоколами байт - ориентированной передачи данных, используемыми теми техническими средствами, в состав которых он входит.

1.3.9. ААПД обеспечивает следующие технические параметры при работе по телефонному каналу связи:

- 1) передача данных: синхронная, асинхронная - устанавливается программным способом;
- 2) скорость передачи данных: при асинхронной передаче до 50, 100, 200, 300, 600, 1200, 2400, 4800, 9600, 19200 бит/с - устанавливается программным способом; при синхронной передаче не выше 19200 бит/с - определяется типом УПС;
- 3) структура символа:

при асинхронной передаче:

число стартовых битов - 1,

число информационных битов 5, 6, 7, 8 - устанавливается программным способом;

число битов контроля по четности, нечетности - 1 или 0 - устанавливается программным способом;

число стоп - битов - 1, 1.5 или 2 - устанавливается программным способом;

при Синхронной передаче:

число информационных битов - 5, 6, 7, 8 - устанавливается программным способом;

число битов контроля по четности, нечетности - 1 или 0 устанавливается программным способом,

1.3.10. ААПД обеспечивает следующие технические параметры при работе по каналам ИРПС:

- 1) передача данных - асинхронная;
- 2) скорость передачи данных до 50, 100, 200, 600, 1200, 2400, 4800, 9600 бит/с - устанавливается программным способом, одинаковая для ИРПС1 - ИРПС 5, для ИРПС6 - одинаковая со скоростью передачи по телефонному каналу связи;
- 3) дальность передачи - до 500 м;

4) структура символа:

число стартовых битов - 1;

число информационных битов - 5, 6, 7, 8 - устанавливается программным способом для каждого канала ИРПС;

число битов контроля по четности или нечетности - 1 или 0 - устанавливается программным способом;

число стоп-битов 1, 1.5, 2- устанавливается программным способом для каждого канала ИРПС.

1.3.11. Электропитание ААПД осуществляется от источников питания технического средства, в состав которого он входит;

(5 ±0.25) В при токе потребления не более 3А;

(12±0.6) В при токе потребления не более 0.4 А;

минус(12±0.6) В при токе потребления не более 0.1 А. Размещение цепей питания по контактам разъемов приведено в табл.5

1.3.12. Элементная база ААПД - интегральные микросхемы серий К155, КР580, К 170, КР 1810, дискретные элементы.

1.3.13. ААПД обеспечивает непрерывную круглосуточную работу.

Таблица 5

Наименование цепи	Номер разъема и контакта
5 В	X1:A1 X1:C1 X2:A32 X2:C32
0 В	X1:A2 X1:C2 X2:A31 X2:C31
12 В Минус 12 В	X1:C3 X1:C30

1.3.14. Средняя наработка ААПД на отказ не менее 40000 ч.

1.3.15. Средняя наработка ААПД на сбой не менее 1-109 символов при нормальных климатических условиях и номинальных значениях питающих напряжений.

1.3.16. Средний срок службы ААПД не менее 10 лет.

1.3.17. Габаритные размеры ААПД - 246x241x16 мм.

1.3.18. Масса ААПД не более 0.5 кг.

1.3.19. ААПД в упаковке для транспортирования выдерживает без повреждений:

1) воздействие температур окружающего воздуха от минус 50оС до плюс 50оС;

2) воздействие относительной влажности окружающего воздуха до 95% при температуре плюс 25оС;

3) воздействие транспортной тряски с ускорением 29.5 м/с² при частоте ударов от 80 до 120 в минуту.

1.3.20. При работе с каналами связи следует предварительно припаять концы линий связи к разъемам, находящимся в ЗИПе (типа РШ2Н-1-5 для подключения линии связи в ранге ИРПС и типа РШ2Н-1-29 для подключения линий связи в ранге С2).

Номенклатура распределения сигналов по контактам разъема РШ2Н-1-29 (С2-100) приведена на рис.1. Распайка разъемов для канала связи в ранге С2 производится в соответствии с рис.2, для каналов связи в ранге ИРПС производится в соответствии с рис.3. Витые пары проводов (длиною до 500 м), используемые для каналов ИРПС, должны соответствовать следующим техническим требованиям:

шаг скрутки, мм, не более 25;

сечение провода, мм², не менее 0,2.

Пример соединения 2-х ПТК по каналам связи в ранге ИРПС, С2 приведён на рис.5, в ранге С2 через модем 1200КН.02 на рис.4.

Для проверки работы ПТК по каналам связи через ААПД к выходному разъёму Х3 ААПД подключить заглушку 6.433.027 из ЗИПа.

ВНИМАНИЕ !

При нарушении заземляющих контактов (вилки, розетки) корпус и “Общая” шина модуля процессора могут находиться под переменным напряжением, передаваемым от сети через входной фильтр БФР, даже при выключенном питании модуля процессора.

В этом случае возможен выход из строя адаптера АПД в момент подключения или отключения кабеля связи (стык С2) к

(от) адаптеру (а) АПД или аппаратуры передачи данных. Для предотвращения этого явления должно быть обеспечено надёжное крепление заземляющего провода к корпусу БФР.

1.4. СОСТАВ ААПД

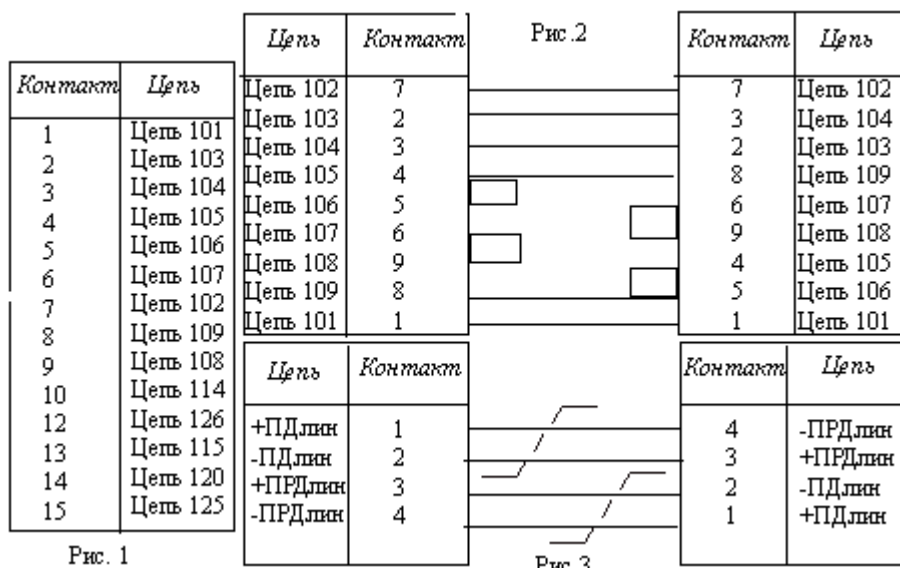


Рис. 1

Рис. 3

1.4.1. В состав изделия входит адаптер АПД - 1 шт.

1.5. УСТРОЙСТВО И РАБОТА ААПД

1.5.1. ААПД обеспечивает взаимодействие технических средств, в состав которых он входит, со стандартными средствами АПД и удаленными устройствами, подключаемыми по рангу ИРПС, и выполняет следующие основные функции :

- 1) одновременную независимую работу по всем каналам последовательной передачи данных ИРПС1 - ИРПС6, С2;
- 2) формирование сигналов прерывания при приеме очередного символа по любому из каналов передачи ИРПС1 - ИРПС6, С2;
- 3) преобразование передаваемых данных из параллельного формата в последовательный с заданными параметрами - скоростью передачи, длиной слова, формированием бита паритета;
- 4) контроль принимаемых последовательных данных по формату и паритету, а также на переполнение приемного буфера;
- 5) коммутацию цепей синхронизации приемо-передатчика стыка С2 - от внутреннего источника в асинхронном режиме или от внешнего (от УПС) - в синхронном;
- 6) трансляцию цепей серии 200 стыка С2 при работе с УАВ;
- 7) взаимодействие с магистралью, заключающееся в дешифрировании адреса обращения к узлу ААПД, формировании внутренних сигналов управления - команд записи и чтения, сигнала готовности для интерфейса и коммутации внутренней магистрали данных;

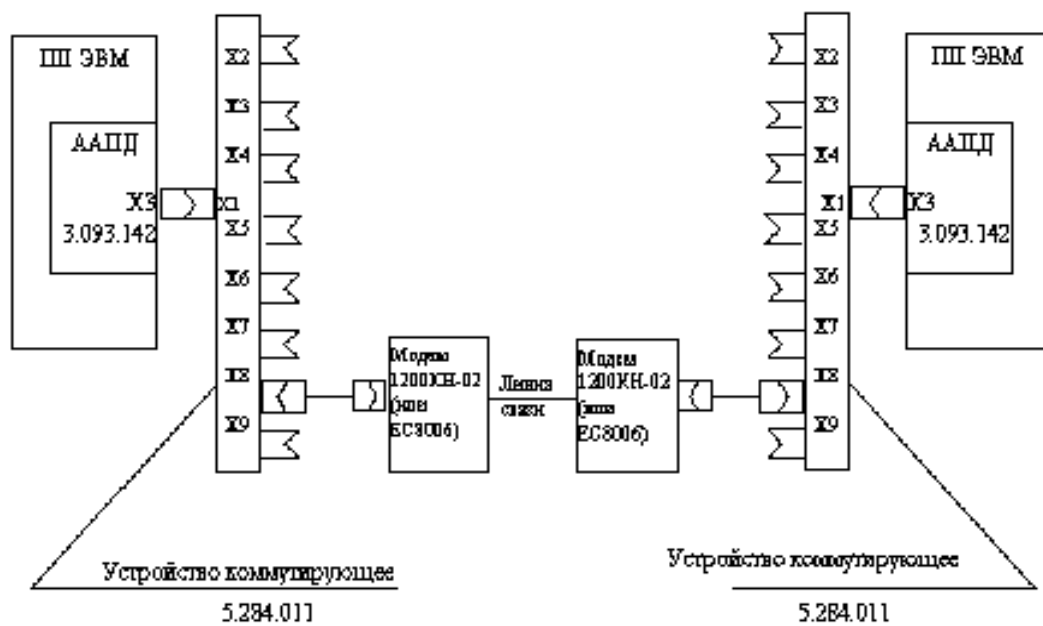


Рисунок 4

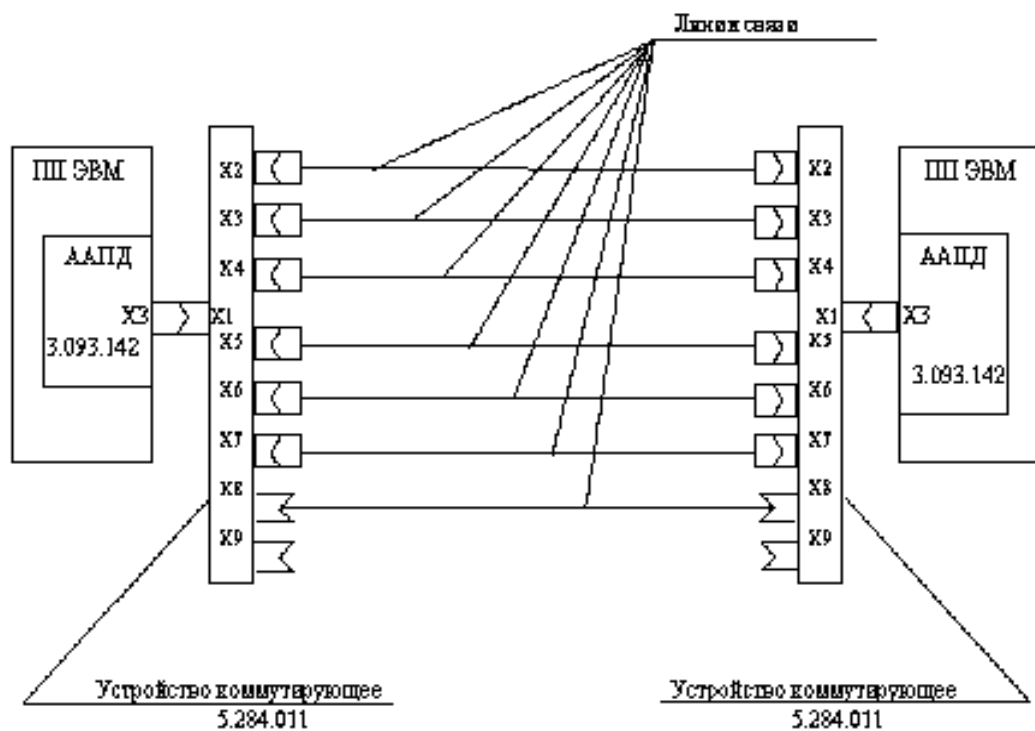


Рисунок 5

- 8) осуществление физического сопряжения с интерфейсом, генераторами и приемниками стыка С2, удаленными устройствами, подключенными по рангу ИРПС;
- 9) тестирование каналов ИРПС и С2, устанавливаемое и осуществляемое программным способом.

1.5.2. С целью обеспечения указанных функций ААПД включает следующие функциональные узлы:

- 1) генератор тактов;
- 2) узел синхронизации SYN1;
- 3) таймер SYN2;
- 4) узел шинных формирователей;
- 5) дешифратор адресов;
- 6) узел управления;
- 7) регистр ввода - вывода IOP;
- 8) узел приема - передачи стыка С2 IOS0;
- 9) узлы приема - передачи ИРПС1 - ИРПС6, IOS1 - IOS6 ;
- 10) узел прерывания INR;
- 11) узел генераторов и приемников стыка С2;
- 12) узел передатчиков и приемников ИРПС;
- 13) узел тестирования.

Структурная схема ААПД приведена на рис.1 Приложения. Схема электрическая принципиальная 3.093.138 ЭЗ приведена на рис.2 Приложения.

1.5.3. Основные функциональные узлы ААПД выполнены на программируемых микросхемах серии КР580:

- 1) узлы приема - передачи ИРПС1 - ИРПС6 и стыка С2 выполнены на микросхемах КР580ВВ51А ;
- 2) узел синхронизации и таймер - на микросхемах КР580ВН53;
- 3) узел прерываний - на микросхеме КР580ВВ59А;
- 4) узел ввода-вывода - на микросхеме КР580ИК55.

Функционирование этих узлов определяется техническими параметрами используемых микросхем и задаваемыми процессором инструкциями режимов (начальной загрузкой).

1.5.4. Адресация узлов ААПД 3.093.138 и их внутренних регистров приведена в табл.6.

1.5.5. Узлы приема-передачи ИРПС функционируют независимо друг от друга.

1.5.5.1. Они осуществляют преобразование данных, поступающих из процессора в параллельном формате, в последовательный, автоматически формируя заданный инструкцией режима формат последовательных данных (бит паритета, длину слова, число стоп-битов).

Готовность узла к передаче в канал связи очередного символа данных выявляется опросом регистра состояния этого узла.

1.5.5.2. При приеме данных из канала связи узлы приема-передачи ИРПС автоматически осуществляют контроль последовательных данных по паритету, наличию стоп-битов, переполнению и преобразование в параллельный формат для передачи в магистраль процессора.

После накопления очередного символа, поступившего из канала связи, узел вырабатывает сигнал готовности принимаемых данных, поступающий на соответствующий вход узла прерывания.

1.5.5.3. В регистре состояния узла при выявлении ошибок в последовательных данных устанавливается соответствующий флаг ошибки.

1.5.6. Узел приема-передачи С2 функционирует

Таблица 6

	Адресные шины A9 A8 A7 A6 A5 A4 A3 A2 A1 A0	Адресуемый узел ААПД	Адресуемый регистр, режим работы узла
2B0	1 0 1 0 1 1 0 0 0 0	IOS0	Рг данных
2B1	1 0 1 0 1 1 0 0 0 1	(C2)	Рг дан., сост.
2B2	1 0 1 0 1 1 0 0 1 0	IOS1	Рг данных
2B3	1 0 1 0 1 1 0 0 1 1	(ИРПС1)	Рг юм., сост.
2B4	1 0 1 0 1 1 0 1 0 0	IOS2	Рг данных
2B5	1 0 1 0 1 1 0 1 0 1	(ИРПС2)	Рг юм., сост.
2B6	1 0 1 0 1 1 0 1 1 0	IOS3	Рг данных
2B7	1 0 1 0 1 1 0 1 1 1	(ИРПС3)	Рг юм., сост.
2B8	1 0 1 0 1 1 1 0 0 0	IOS4	Рг данных
2B9	1 0 1 0 1 1 1 0 0 1	(ИРПС4)	Рг юм., сост.
2BA	1 0 1 0 1 1 1 0 1 0	IOS5	Рг данных
2BB	1 0 1 0 1 1 1 0 1 1	(ИРПС5)	Рг юм., сост.
2BC	1 0 1 0 1 1 1 1 0 0	IOS6	Рг данных
2BD	1 0 1 0 1 1 1 1 0 1	(ИРПС6)	Рг юм., сост.
2BE	1 0 1 0 1 1 1 1 1 0	INR	Работа в соот. с инстр. по программир. KP580BH59A
2BF	1 0 1 1 1 1 1 1 1 1		
278	1 0 0 1 1 1 1 0 0 0	SYN1	Счетчик 0
279	1 0 0 1 1 1 1 0 0 1		Счетчик 1
27A	1 0 0 1 1 1 1 0 1 0		Счетчик 2
27B	1 0 0 1 1 1 1 0 1 1		Рг управления
27C	1 0 0 1 1 1 1 1 0 0	SYN2	Счетчик 0
27D	1 0 0 1 1 1 1 1 0 1		Счетчик 1
27E	1 0 0 1 1 1 1 1 1 0		Счетчик 2
27F	1 0 0 1 1 1 1 1 1 1		Рг управления
2F8	1 0 0 1 1 1 1 0 0 0	IOF	Порт А
2F9	1 0 0 1 1 1 1 0 0 1		Порт В
2FA	1 0 0 1 1 1 1 0 1 0		Порт С
2FB	1 0 0 1 1 1 1 0 1 1		Рг управления

полностью аналогично узлам приема-передачи ИРПС, и, кроме того, может быть запрограммирован для работы в синхронном режиме (при использовании синхронных модемов).

1.5.6.1. При поступлении из канала связи символа данных узел приема-передачи С2 также вырабатывает сигнал готовности принимаемых данных, поступающий на вход узла прерывания.

1.5.6.2. При передаче данных в канал узла связи готовность узла выявляется опросом регистра состояния узла.

1.5.7. Синхронизация данных в узлах приема-передачи ИРПС, а также узла приема-передачи С2 при работе в асинхронном режиме осуществляется при помощи узла синхронизации SYN1.

1.5.7.1. На входы независимых программируемых счетчиков этого узла поступает тактовая последовательность с частотой $f=1228,8\text{ кГц}$ и скважностью $Q=2$ со счетного триггера, на котором осуществляется деление на 2 частоты тактового генератора, стабилизированного кварцевым резонатором $2457,6\text{ кГц}$.

1.5.7.2. В зависимости от условий синхронизации IOS узлов ИРПС и С2 (X1, X16 или X64) и требуемой скорости приема-передачи последовательных данных в каналах связи счетчикам программным способом задается коэффициент деления входной частоты. Рекомендуемый коэффициент деления определяется условием синхронизации IOS - X16.

1.5.7.3. На счетчике Сч1 формируется частота синхронизации данных для узлов ИРПС1- ИРПС3, на счетчике Сч0 - передатчиков узлов ИРПС6 и С2, на Сч2 -приемников и передатчиков ИРПС4 и ИРПС5.

1.5.7.4. Счетчики программируются в режиме 3 (генератор меандра). Коэффициенты деления счетчиков выбираются в соответствии с указаниями табл.7.

1.5.8. ААПД вырабатывает неуправляемую последовательность t и управляемую тактовую последовательность ТА для получения временных интервалов в соответствии с командами управления, вырабатываемыми программным способом в соответствии с п. 1.6.7 настоящего ТО.

1.5.8.1. Тактовая последовательность формируется путем деления входной частоты $1228,8\text{ кГц}$. Деление осуществляется на Сч0 таймера SYN2. Коэффициент деления счетчика $m1$. На счетчике Сч1 таймера SYN2 осуществляется первая ступень деления для получения требуемой частоты ТА, коэффициент деления $m2$.

1.5.8.2. На счетчике Сч2 таймера SYN2 при помощи задания программным способом коэффициента деления осуществляется вторая ступень деления тактовой последовательности ТА. Коэффициент деления $m3$.

1.5.8.3. Сч2 таймера SYN2 может быть использован в режиме формирования временного интервала - его выход поступает на соответствующий вход узла прерывания INR.

1.5.8.4. Программирование таймера SYN2 (режим работы и коэффициенты деления счетчиков $m1 - m3$) целиком определяется требованиями технического средства, в состав которого входит адаптер.

1.5.9. Режим тестирования заключается в коммутации выхода последовательных передаваемых данных каждого из IOS узлов ИРПС1 - ИРПС6 и С2 на вход последовательных принимаемых данных. Режим задается программным способом, что позволяет проверить работоспособность приема-передатчиков без остановки работы технического средства в целом.

Таблица 7

Скорость передачи данных, бит/сек	Коэффициент деления			
	десятичный			двоичный
	Условие синхронизации IOS			Условие синхронизации IOS
	X1	X16	X64	X16
9600	128	8	-	8
4800	256	16	-	10
2400	512	32	8	20
1200	1024	64	16	40
600	2048	128	32	50
300	4096	256	64	60
200	6144	384	96	180
100	12288	768	192	300
50	24576	1536	384	600

1.6. УСТРОЙСТВО И РАБОТА СОСТАВНЫХ ЧАСТЕЙ ААПД

1.6.1. Генератор тактов, собранный на элементах D15.1 и D15.2, резисторах R12 - R15 и конденсаторе C1, стабилизирован кварцевым резонатором B1.

Генератор выдает тактовую последовательность со следующими параметрами:

- 1) частота $f=2457,6$ кГц;
- 2) скважность $Q=2$;
- 3) относительная нестабильность не более $1 \cdot 10^{-4}$.

С выхода генератора сигналы поступают на счетный триггер D16.1, на котором происходит деление частоты на 2. С его выхода тактовая последовательность с частотой $f=1228,8$ кГц поступает на входы узла синхронизации.

1.6.2. Узел шинных формирователей реализован на микросхемах D3, D14 - D14 KP580BA86. Узел формирует внутреннюю двунаправленную магистраль данных и осуществляет физическое сопряжение ее с магистралью процессора.

Шинный формирователь D12 формирует шину данных для узлов IOS1, IOS4 - IOS6, D13 - для узлов IOP, INR и IOS0, D14 - для узлов SYN1, SYN2, IOS2, IOS3. По сигналам из дешифратора адресов на один из формирователей подается разрешающий сигнал $E_{\blacklozenge}$. Направление прохождения информации определяет задаваемая процессором операция - запись () или чтения ().

1.6.3. Дешифратор адресов построен на элементах D4, D5, D6, D8, D9.1.

В зависимости от комбинации сигналов на адресных шинах дешифратор вырабатывает сигналы выбора узлов адаптера (в соответствии с табл.8 и 9). D4 представляет собой запрограммированную микросхему KP556IPT5 и выдает в зависимости от комбинации адресных и управляющих сигналов на ее адресных и управляющих входах комбинацию сигналов на выходах. Сигналы на выходе D4 определяют, какой из дешифраторов D5, D6 должен быть активизирован, кроме того D4 транслирует сигналы двух младших разрядов адреса и сигналы A0 - A1.

При наличии сигнала на входе D4 он блокируется. При наличии сигнала INT на входе блокируется все выходы D4, кроме выходов CAS. Выбор одной из внутренних магистралей данных осуществляется на микросхемах D8 и D9.1.

1.6.4. Узел управления включает триггер D2 и отдельные части микросхем D7, D10, D11. На триггерах D2 осуществляется развязка сигналов управления чтением, записью, а также

Таблица 8.

Разряд шины данных	Наименование цепи, режим порта		
	Порт А, режим 0, ввод	Порт В, режим 0, вывод	Порт С, режим 0 D7-D4 - вывод D3-D0 - ввод
D7	213	202	Режим тестирования
D6	203	211	Управление таймером
D5	204	206	-
D4	205	207	-
D3	210	208	107
D2	125	209	106
D1	104	126	-
D0	109	Вид синхронизации синхр./асинхр.	-

Таблица 9.

Наименование цепи	Уровень сигнала	Разряд шины данных	Режим
Вид синхронизации синхронная/асинхронная	0 1	$\overline{D0}$	Асинхронная передача данных Синхронная передача данных
Управление таймером	0 1	$\overline{D0}$	Запрет счета, установка таймера Разрешение счета
$\overline{I26}$	0 1	$\overline{D1}$	Верхняя частота передачи Нижняя частота передачи
Режим тестирования	0 1	$\overline{D7}$	Разрешение «Автономная работа» Запрещение «Неавтономная работа»

начальной установки. С выхода D2:01 через D7.1 выдается сигнал записи на все узлы ААПД, с выхода D2:14 через D7.2 - сигнал чтения.

На D10.1 и D11.1, D11.2 вырабатывается сигнал готовности. Сигнал готовности вырабатывается при наличии одного из следующих трех условий:

сигнал записи в один из узлов ААПД (WR* ADR);

сигнал чтения одного из узлов ААПД (RD*ADR);

наличие сигналов подтверждения запроса прерывания и разрешения чтения кода прерывания (INTA*RA).

На элементах D11.1 и D11.2 осуществляется требуемая задержка сигнала готовности, величина которой составляет 600 - 700 нс.

1.6.5. Регистр ввода-вывода IOP выполнен на микросхеме D19 KP580ИК55, содержащей три независимых порта А, В и С. Регистр используется для ввода в процессор и вывода на генераторы стыка С2 ряда сигналов серий 100 и 200 управления режимом тестирования и таймером. Размещение сигналов по портам и

их распределение по разрядам шины данных (при обмене с процессором) приведено в табл.8.

Выбор режима в зависимости от сигналов в цепях управления определяется таблицей 9.

1.6.6. Узлы приёма-передачи выполнены на микросхемах KP580BB51A.

Узел приёма-передачи C2 - на микросхеме D20 (IOS0);

Узел приёма-передачи ИРПС1 - D21 (IOS1);

ИРПС2 - D22 (IOS2);

ИРПС3 - D23 (IOS3);

ИРПС4 - D24 (IOS4);

ИРПС5 - D25 (IOS5);

ИРПС6 - D26 (IOS6);

1.6.6.1. При программировании приёмо-передатчиков должно быть учтено следующее:

1) режим передачи IOS0 - в зависимости от требований к адаптеру синхронный или асинхронный;

Режим передачи IOS1 - IOS6 - асинхронный;

2) структура символа - в зависимости от требований к адаптеру;

3) для IOS в синхронном режиме - число синхросимволов в зависимости от требований к адаптеру;

4) кратность синхронизации для IOS1-IOS6 и для IOS0 в асинхронном режиме должна быть согласована с коэффициентом деления счётчиков узла синхронизации. Предпочтительно $16 * V$, где V- скорость приёма-передачи последовательных данных;

5) для IOS0 в синхронном режиме синхронизация - внешняя, SYNDET- в положении “Выход”.

1.6.6.2. Для обеспечения функционирования в узлах IOS1- IOS6 выходы RTS подключены ко входам CTS, а выходы DTR- ко входам DSR.

1.6.6.3. В режиме передачи для всех IOS - передача очередного байта данных- по результатам опроса регистра состояния IOS.

В режиме приёма - приём байта как по результатам опроса состояния IOS, так и по запросам прерывания от узла прерывания INR.

1.6.7. Узел прерывания INR выполнен на микросхеме D27 KP1810BH59A и обеспечивает функционирование в двух режимах:

1) режим работы по запросам прерывания;

2) режим работы по результатам опроса.

Для обеспечения работы по запросам прерывания к узлу прерывания подключены цепи каскадирования CAS0-CAS2.

Подключение сигналов прерывания ко входам узла прерывания приведено в таблице 10.

1.6.8. Узел генераторов и приёмников стыка C2 осуществляет сопряжение сигналов ТТЛ с биполярными сигналами по цепям стыка C2.

Генераторы стыка выполнены на микросхемах D32- D36 K170АП2, приёмники- D38- D40 K170УП2.

1.6.9. Передатчики ИРПС Пд1- Пд6 осуществляют физическое преобразование сигналов ТТЛ в токовые посылки (20 мА токовая петля), а приёмники Пр1- Пр6- обратное преобразование.

Передатчик ИРПС включает токовый ключ на транзисторе V2, являющемся регулятором тока, транзистор V1 включен в цепь регулирующей обратной связи. Диоды V3, V4 выполняют защитные функции.

Приёмник ИРПС включает транзисторный оптрон V3,

Таблица 10.

<i>Выход INR</i>	<i>Подключаемый сигнал</i>	<i>Узел, источник прерывания</i>
RQIN R0	TA	Таймер, SYN2
RQIN R1	ERD	ИРПС1, IOS1
RQIN R2	ERD	ИРПС2, IOS2
RQIN R3	ERD	ИРПС3, IOS3
RQIN R4	ERD	ИРПС4, IOS4
RQIN R5	ERD	ИРПС5, IOS5
RQIN R6	ERD	ИРПС6, IOS6
RQIN R7	ERD	Стык С2, IOS0

выполняющий функцию гальванической развязки. Транзистор V2 усиливает сигнал с выхода оптрона, диод V1 осуществляет защитную функцию.

1.6.10. Узел тестирования построен на шинных формирователях D28 и D29 и коммутаторах D30 и D31. В режиме тестирования сигналы коммутируются в соответствии с таблицей 11.

1.7. КОНСТРУКЦИЯ ААПД.

1.7.1. Конструктивно ААПД выполнен на типовой плате “Е2” с тремя разъёмами X1-X3 СНП 59-64/94х11 В-23-2-В.

1.7.2. Вилки X1 и X2 используются для подключения ААПД к магистрали процессора, к локальному или системному интерфейсу (в зависимости от варианта исполнения ААПД).

Таблица 11.

<i>Узел, выходной сигнал</i>	<i>Сигнал, источник сигнала</i>
IOS0, RCD (цепь 104)	TFD (цепь 103), IOS0
IOP, PAO (цепь 109)	DTR (цепь 108), IOS0
IOS0, CTS (цепь 106)	RTS (цепь 105), IOS0
IOS0, DSR (цепь 107)	DTR (цепь 108), IOS0
IOS1, RCD (ПрД1)	TFD (ПД1), IOS1
IOS2, RCD (ПрД2)	TFD (ПД2), IOS2
IOS3, RCD (ПрД3)	TFD (ПД3), IOS3
IOS4, RCD (ПрД4)	TFD (ПД4), IOS4
IOS5, RCD (ПрД5)	TFD (ПД5), IOS5
IOS6, RCD (ПрД6)	TFD (ПД6), IOS6

1.7.3. Вилка X3 используется для подключения к ААПД удалённых устройств по каналам ИРПС1-ИРПС6, а также УПС и УАВ по цепям серии 100 и 200 стыка С2.

1.7.4. Вилка X3 размещена в одном конструктиве с планкой, снабжённой экстракторами. Планка имеет гнезда для фиксации ответной части разъёма X3.

1.7.5. Перечисленные выше каналы связи могут быть подключены к ААПД либо непосредственно распайкой на одной розетке, сопрягаемой с X3, либо каждый своим разъёмом к устройству коммутирующему 5.280.003.

1.8. МАРКИРОВКА.

Маркирование ААПД осуществляется наклейкой на рамку съёмника шильдика с надписью “ААПД”.

1.9. ЭКСПЛУАТАЦИЯ ААПД.

Эксплуатация ААПД осуществляется в соответствии с 1.320.003-04 ТО (п. 9.7) и 1.320.003 ИЭ (п. 5.8).

1.10. УКАЗАНИЕ МЕР БЕЗОПАСНОСТИ.

Персонал, производящий обслуживание и ремонт ААПД, должен быть аттестован на III квалификационную группу по технике безопасности и иметь допуск на обслуживание электроустановок до 1000 В.

Ремонт, монтажные работы на плате ААПД, подключение и отключение жгутов производить только при выключенном напряжении питания.

1.11. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ.

Техническое обслуживание ААПД производится в составе комплекса, в котором ААПД используется. После выполнения технического обслуживания ААПД необходимо произвести проверку работоспособности выполнением тест-программы проверки ААПД.

1.12. ПРИЛОЖЕНИЯ.

В приложении представлены:
сборочный чертеж ААПД (3.093.142 СБ);
структурная схема ААПД (Рис. 1);
схема электрическая принципиальная (3.193.138 ЭЗ)0



Центральный архив УРСР







III

15 ПИ4	ХЗ:С14
16 ПИ5	ХЗ:С6
160 ПИ1	ХЗ:А1
161 ПИ1	ХЗ:А2
172 ПИ1	ХЗ:С1
173 ПИ1	ХЗ:С2
162 ПИ2	ХЗ:А4
163 ПИ2	ХЗ:А6
174 ПИ2	ХЗ:С4
175 ПИ2	ХЗ:С5
164 ПИ3	ХЗ:А7
165 ПИ3	ХЗ:А8
176 ПИ3	ХЗ:С7
177 ПИ3	ХЗ:С8
166 ПИ4	ХЗ:А10
167 ПИ4	ХЗ:А11
178 ПИ4	ХЗ:С10
179 ПИ4	ХЗ:С11
168 ПИ5	ХЗ:А13
169 ПИ5	ХЗ:А14
180 ПИ5	ХЗ:С13
181 ПИ5	ХЗ:С14
170 ПИ6	ХЗ:А16
171 ПИ6	ХЗ:А17
182 ПИ6	ХЗ:С16
183 ПИ6	ХЗ:С19
С. Цена 182	ХЗ:А19
184 Цена 183	ХЗ:С23
185 Цена 184	ХЗ:А28
186 Цена 185	ХЗ:С24
187 Цена 186	ХЗ:А27
188 Цена 187	ХЗ:А26
189 Цена 188	ХЗ:С26
190 Цена 189	ХЗ:С25
191 Цена 114	ХЗ:С27
192 Цена 115	ХЗ:А31
193 Цена 120	ХЗ:А23
194 Цена 125	ХЗ:С31
195 Цена 126	ХЗ:С21
С. Цена 281	ХЗ:А32
202 Цена 282	ХЗ:А30
203 Цена 283	ХЗ:С30
204 Цена 284	ХЗ:А29
205 Цена 285	ХЗ:С29
206 Цена 286	ХЗ:С18
207 Цена 287	ХЗ:С12
208 Цена 288	ХЗ:С15
209 Цена 289	ХЗ:С3
210 Цена 218	ХЗ:С32
211 Цена 211	ХЗ:С3
213 Цена 213	ХЗ:С28
Цена 181	ХЗ:С20
Цена 212	

Поз. обознач.	Наименование	Кол.	Примечание
А7, А8	МЛТ-0.125-1кОм+10%	2	
А9	МЛТ-0.125-10кОм+10%	1	
А10, А16	МЛТ-0.125-1кОм+10%	1	
А17	МЛТ-0.5-300 Ом+10%	1	
А18, А23	МЛТ-0.125-200 Ом+10%	6	
А24, А25	МЛТ-0.125-1кОм+10%	1	
U1	Диод КД521А	1	
П1-П6	Передачик КРПС	6	
	<u>Резисторы</u>		
А1	МЛТ-0.125-33 Ом+5%	1	
А2	МЛТ-0.125-390 Ом+10%	1	
А3	МЛТ-0.125-1.5кОм+10%	1	
U1, U2	Транзистор КТ3107Б	2	
U3, U4	Диод КД521А	2	
Пр1-Пр6	Приемник ИРПС	6	
	<u>Резисторы</u>		
А1	МЛТ-0.125-100 Ом+5%	1	
А2	МЛТ-0.125-1.1кОм+10%	1	
U1	Диод КД521А	1	
U2	Оптран А0Д 129А	1	
U4	Транзистор КТ 3102Б	1	Доп.зан КТ3102БМ
Х1, Х2	Вилка СМП59-96/34 11В-23-2В	2	
Х3	Вилка СМП59-64/34 11В-23-2В	1	

Подключение цепей питания микросхем

Поз. обозначение микросхем	А (+12В)	В (+5В)	С (0В/ш.0)	Д (-12В)
	контакты			
D1, D3, D4, D7, D12-D14, D28, D29		20	10	
D2		5	12	
D17, D18		24	12	
D8-D11, D15, D16, D37, D43		14	7	
D20-D26		26	4	
D27		28	14	
D5, D6, D38, D41, D42		16	8	
D32-D36	8		4	5
D39-D40	16		8	
D19		26	7	

Поз. обознач.	Наименование	Кол.	Примечание
В1	Резонатор РК1706В-7АП-2457.6кГц	1	
	<u>Конденсаторы</u>		
С1	К10-10-Н90-0.015нФ+10%	1	
С2-С6	К10-17-3Г-М47-1000пФ+20%	5	
С7-С11	К50-35-63В-10нФ	5	
С12, С13	К10-П-3Г-Н90-0.060нФ	2	
С14, С39	К10-17-3Д-Н90-0.1нФ	26	
С40	К10-10-М75-33пФ+10%	1	
	<u>Микросхемы</u>		
D1	КР500ВА06	1	
D2	К155ТМ7	1	
D3	КР500ВА07	1	
D4	К555АП3	1	
D5, D6	К555ИД4	2	
D7	КР500ВА07	1	
D8, D9	К555ЛМ6	2	
D10	К155ЛМ1	1	
D11	К155ЛМВ	1	
D12, D14	КР500ВА06	3	
D15	К166ЛН3	1	
D16	К555ТМ2	1	
D17, D18	КР500ВН53	2	
D19	КР500ВВ55А	1	
D20, D28	КР500ВВ51А	7	
D27	КР1010ВН59А	1	
D28, D29	КР500ВА06	2	
D30, D31	К555КП2	2	
D32, D38	К170АП2	5	
D37	К155ЛН3	1	
D38, D40	К170УП2	3	
D41	К155ЛП11	1	
D42	К555КП11	1	
D43	К555ЛА4	1	
	<u>Резисторы</u>		
Р1	МЛТ-0.125-1кОм+10%	1	
Р2	МЛТ-0.125-10кОм+10%	1	
Р3	МЛТ-0.125-1кОм+5%	1	
Р4	МЛТ-0.125-430 Ом+5%	1	
Р5	МЛТ-0.125-1кОм+6%	1	
Р6	МЛТ-0.125-430 Ом+5%	1	

Рис.2. ААПД. Схем электрическая принципиальная 3.093.183 ЭЗ (лист 4)

Лабораторная работа 1 (описание)

**2. «БИС программируемого таймера
КР1810ВИ54 (і8254) »**

Содержит описание лабораторной работы «БИС программируемого таймера КР1810ВИ54 (i8284)», проводимой в рамках курса «Адаптеры и контроллеры ЭВМ» с целью изучения принципов работы и программирования программируемого таймера КР1810ВИ54(i8254) в составе подсистемы таймера IBM/PC-подобных ПЭВМ. Предназначено для студентов дневных и вечерних групп.

2.1. ЦЕЛЬ РАБОТЫ :

Целью лабораторной работы «БИС программируемого таймера КР1810ВИ54» является изучение режимов работы программируемого таймера КР1810ВИ54 (i 8254) в составе подсистемы таймера ПЭВМ «Искра 1031», а также приемов его программной настройки на различные режимы работы.

2.2. ТЕХНИЧЕСКИЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ В ЛАБОРАТОРНОЙ РАБОТЕ :

В лабораторной работе используется :

- ПЭВМ «Искра 1031» (IBM PC/AT)
- осциллограф С1-81

2.3. НАЗНАЧЕНИЕ ПРОГРАММИРУЕМОГО ТАЙМЕРА КР1810ВИ54 (i 8254) :

Программируемый таймер представляет собой микросхему, работающую независимо от процессора и предназначен для выполнения разных функций, связанных с отсчетом временных интервалов. Таймер может быть запрограммирован для генерации отдельных импульсов, импульсов с заданной частотой или генерации задержанных импульсов.

2.4. ФУНКЦИОНИРОВАНИЕ КР1810ВИ54 (i 8254) :

Таймер может работать в 6 режимах :

Примеры временных диаграмм работы программируемого таймера в различных режимах можно посмотреть в приложении 2.8.4.

1) Режим 0 - «Прерывание таймера» (программируемая задержка).

В этом режиме низкий уровень на выходе OUT устанавливается сразу после загрузки управляющего слова. Загрузка константы не оказывает влияние на этот выход. Счет разрешается положительным сигналом на входе GATA. Изменение содержимого счетчика SE осуществляется по первому срезу сигнала CLK, причем, по первому срезу происходит загрузка константы из CR в SE и только второй тактовый сигнал начинает счет. После отсчета загруженного числа, на выходе OUT устанавливается «1». Таким образом, сигнал $OUT=0$ удерживается на время равное $(n+1)$ тактовых импульсов, где n - загруженная константа. Если во время счета снять сигнал GATA, то счет приостановится, но содержимое счетчика таймера сохранится. Новый положительный сигнал GATA вызывает продолжение счета без перезагрузки SE содержимым CR. Загрузка новой константы во время счета приводит :

- при записи младшего байта - к остановке текущего счета;
- при записи старшего байта - к запуску нового цикла счета.

Контроль счета (выполнение команд CLC и RBC) в этом режиме возможен только после выполнения хотя бы одного цикла счета.

2) Режим 1 - «Программируемый ждущий одновибратор».

В этом режиме на выходе OUT формируется сигнал низкого уровня длительностью $T = TCLK * N$, где TCLK - период тактовых импульсов, N - константа пересчета. Низкий уровень сигнала OUT устанавливается по переднему фронту сигнала GATA и снимается по окончании счета. Этот режим с перезапуском, т.е. по каждому фронту сигнала GATA регистр SE перезагружается содержимым CR. По фронту первого сигнала GATA флаг обновления устанавливается в «0». Если во время счета в программируемый таймер загружается новая константа, то она устанавливает флаг обновления в «1», но не влияет на текущий счет. Новый счет начинается только по фронту следующего сигнала GATA. Выполнение команд CLC и RBC возможно только после выполнения хотя бы одного цикла счета.

3) Режим 2 - «Импульсный генератор частоты».

Канал работает как делитель входной частоты CLK/NCONST. Сразу же после загрузки управляющего слова, на выходе OUT устанавливается единичный сигнал. При GATA=1 на выходе OUT, с частотой равной FCLK/N, устанавливается нулевой уровень на время, равное одному периоду TCLK. Этот режим с автозагрузкой, т.е. после окончания цикла счета, SE автоматически перезагружается из CR и счет повторяется. Перезагрузка канала новой константой не влияет на текущий счет. Новый счет начинается по окончании предыдущего. При GATA=0, на выходе OUT устанавливается «1» и счет прекращается. При восстановлении GATA=1, счет продолжается, что позволяет синхронизировать работу канала с внешними событиями. Выполнение команд CLC и RBC возможно после окончания двух циклов счета.

4) Режим 3 - «Генератор импульсов со скважностью 2».

Этот режим аналогичен режиму 2, но на выходе OUT формируются импульсы с длительностью полупериода :

$$\begin{aligned} &= TCLK * N / 2, \quad \text{при четном } N; \\ &= TCLK * (N+1) / 2, \quad \text{при нечетном } N - \text{ для} \\ &\text{положительного полупериода;} \\ &= TCLK * (N-1) / 2, \quad \text{при нечетном } N - \text{ для} \\ &\text{отрицательного полупериода.} \end{aligned}$$

В этом режиме канал может работать только с константой $N > 3$.

5) Режим 4 - «Программно запускаемый одновибратор».

По окончании счета на выходе OUT устанавливается низкий уровень на время одного периода сигнала CLK. Высокий уровень на выходе OUT устанавливается сразу же после загрузки управляющего слова. GATA=1 - разрешает счет, причем, первым тактовым сигналом происходит загрузка счетчика таймера SE константой из CR, а последующие сигналы CLK осуществляют счет. Таким образом, сигнал низкого уровня на выходе OUT устанавливается через (N+1) тактовых периодов. Если во время счета снимается сигнал GATA, то счет приостанавливается. Текущее значение SE сохраняется. Новый положительный сигнал GATA вызывает продолжение счета. Это одnorазовый режим выполнения функции. Загрузка новой константы во время счета приводит :

- при записи младшего байта - к остановке текущего счета;
- при записи старшего байта - к запуску нового цикла счета.

Выполнение команд CLC и RBC возможно только после выполнения хотя бы одного цикла счета.

б) Режим 5 - «Аппаратно запускаемый одновибратор».

Этот режим аналогичен режиму 4 по способу формирования сигнала на выходе OUT и режиму 1 по действию сигнала GATA. На выходе OUT устанавливается сигнал низкого уровня на время одного периода тактового импульса TCLK после отсчета загруженной константы. Загрузка константы в СЕ из CR осуществляется по фронту сигнала GATA, причем, первый фронт GATA устанавливает флаг обновления в «0». Если во время счета в канал загружается новая константа, то флаг обновления восстанавливается, но не оказывает влияние на текущий счет. Новый счет начинается по фронту следующего сигнала GATA. Выполнение команд CLC и RBC возможно только после выполнения хотя бы одного цикла счета.

Замечания по режимам :

Из описания режимов работы программируемого таймера видно, что таймер может реализовывать все основные время задающие функции, широко используемые в цифровой технике. При конструктивной совместимости таймеров КР580ВИ53 и КР1810ВИ54, последний кроме улучшенных частотных характеристик обладает более широкими функциональными возможностями.

2.5. ПОДГОТОВКА К ЛАБОРАТОРНОЙ РАБОТЕ .

Ознакомиться:

☐ с описанием лабораторной работы «БИС программируемого таймера КР1810ВИ54 (i 8254)»

☐ с описанием работы микросхемы КР1810ВИ54 (i 8254),

☐ с командами отладчика AFD,

☐ примерами программ (см. Приложения 2.8.2. и 2.8.3. к лабораторной работе).

Изучить режимы работы программируемого таймера КР1810ВИ54 (i 8254).

2.6. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ :

2.6.1 Пользуясь командами О и I отладчика AFD, установить канал 2 таймера в режим 3 (генератор меандра) и занести в регистр CR канала константу 1010h. Затем через порт 61 подать «1» на вход GATA 2 программируемого таймера и наблюдать сигнал на выходе OUT 2 таймера по осциллографу. Измерить параметры сигнала и сравнить с расчетными. Далее подать через порт 61 бит разрешения выдачи сигнала на встроенный динамик (определить тональность сигнала).

Включение звука : 4В - в порт 61;

Выключение звука : 48 - в порт 61.

2.6.2 Написать на ассемблере программу, реализующую периодическое изменение тональности звука (период около 1 секунды). Набрать и запустить на счет в среде отладчика AFD.

2.6.3 Исследовать работу канала 2 программируемого таймера в остальных режимах.

2.6.4 Оформить отчет по результатам лабораторной работы.

2.7. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА :

- Справочник «Аппаратура персональных компьютеров и ее программирование IBM PC XT/AT и PS/2 », Москва, «Радио и связь», 1995г.
- Лекции по курсу «Адаптеры и контроллеры ЭВМ».

2. 8. ПРИЛОЖЕНИЯ :

2.8.1. ОПИСАНИЕ КР1810ВИ54 (i 8254) :

СПРАВОЧНЫЕ МАТЕРИАЛЫ ПО ПРОГРАММИРУЕМОМУ ТАЙМЕРУ КР1810ВИ54

Общие характеристики :

Программируемый таймер предназначен для генерации время задающих функций, программно управляемых временных задержек с возможностью программного контроля их управления. Тактовая частота программируемого таймера до 8 МГц.

Конструктивно этот программируемый таймер совместим с К580ВИ53 (i8253).

Отличается повышенным быстродействием и расширенными функциональными возможностями.

Программируемый таймер включает в себя :

- Буфер шины данных BD, логические схемы управления чтением/записью;
- Дешифратор DS, с помощью которого выбирается один из 3 независимых каналов, каждый из которых может работать в одном из 6 режимов для двоичного и двоично-десятичного счета, либо формирует признак загрузки управляющего слова или команды;
- 3 информационных канала COUNT 0-2, реализующих запрограммированную функцию.

Каждый канал включает в себя :

- 16 разрядный буферный регистр CL0, который служит для запоминания и хранения мгновенного значения счетчика CE0, которое в любое время может быть записано командой «защелка» или «чтение состояния канала». После выполнения этих команд, содержимое CL0 может быть считано в ЦП без остановки счета в регистре CE0.
- 16 разрядный счетчик таймера CE0, работающий в режиме вычитания. Изменение содержимого счетчика CE0 осуществляется по срезу CLK при GATA=1.
- 16 разрядный регистр констант пересчета CR0, служащий для хранения констант пересчета. Содержимое CR0 загружается в CE0 для счета, в зависимости от запрограммированного режима.
- 8 разрядный регистр состояния RS0, содержимое которого можно прочитать с помощью команды «чтение состояния канала» RBC.
- 8 разрядный регистр управляющего слова RCW0, предназначенный для хранения информации, определяющей режим работы канала.

Управляющее слово загружается в RCW0 по команде OUT с адресом, формирующем на входах A0 и A1 код 11. Выбор конкретного канала осуществляется с помощью двух старших разрядов самого управляющего слова.

Схема управляющей логики канала CL, осуществляющая управление входом/выходом счетчика таймера, в зависимости от запрограммированного режима.

По правилам загрузки счетчика таймера CE0, содержимым регистра CR0, все 6 режимов работы программируемого таймера, можно разделить на 3 группы :

1) Режимы 0 и 4 - режимы однократного выполнения функции.

Константы из CR0 передаются в CE0 по первому тактовому сигналу CLK при GATA=1 и разрешении работы канала. С приходом последующих CLK, происходит уменьшение CE0. Если во время счета на вход GATA подать 0, то это приведет к останову счета.

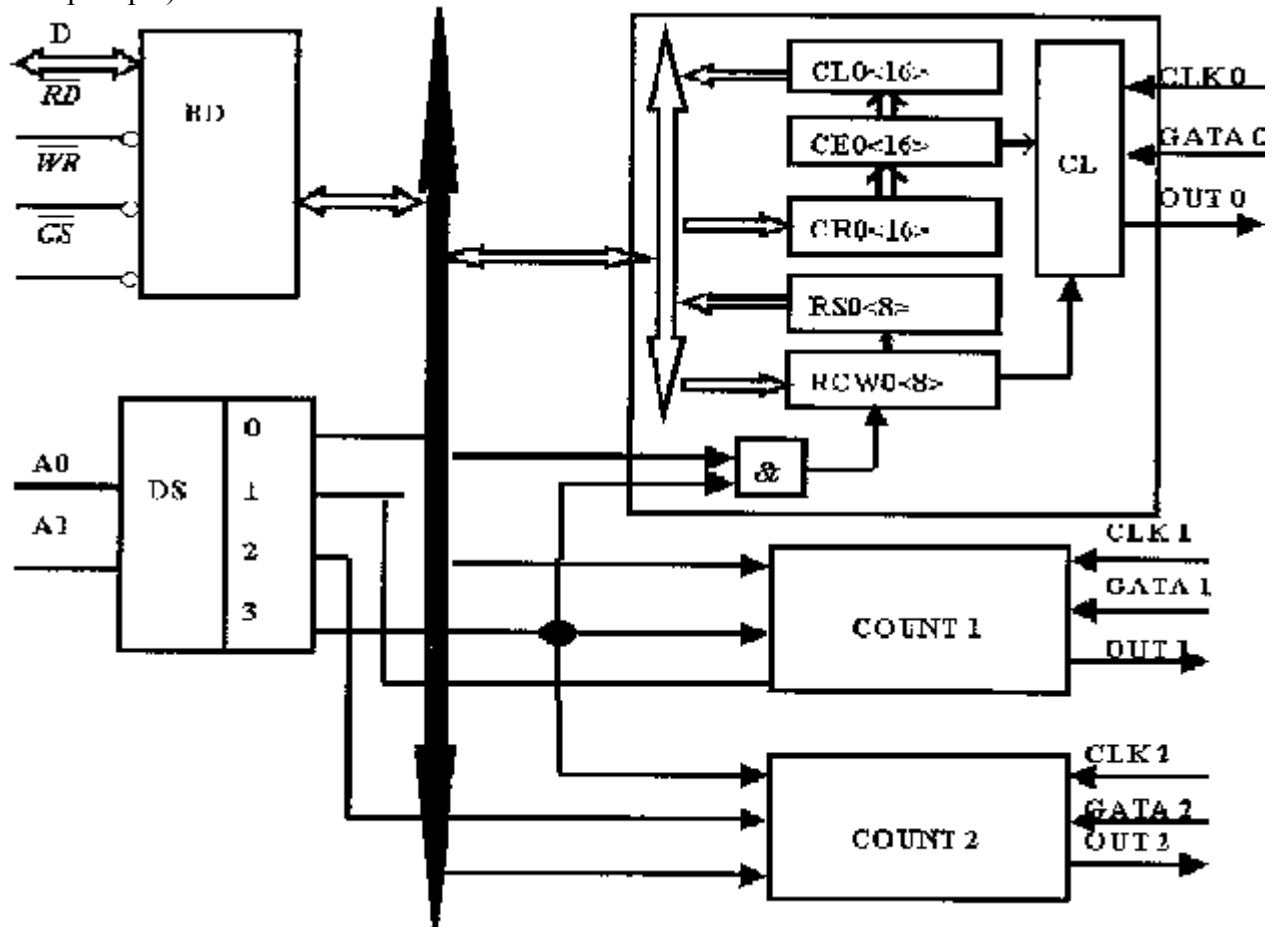
Новый положительный сигнал на GATA не вызывает перезагрузки счетчика, а только разрешает продолжение счета. По окончании счета выполнение действий заканчивается. При необходимости повторения функции требуется новая загрузка константы.

2) Режимы 1 и 5 - режимы с перезапуском.

Для них характерна возможность повторения запрограммированных функций без нового программирования. Загружаемая константа при этом сохраняется в CR0, а ее передача в CE0 осуществляется по фронту сигнала GATA, независимо от завершения счета.

3) Режимы 2 и 3 - режимы автозагрузки.

Загрузка CE0 содержимым CR0 осуществляется автоматически при выполнении условий счета. Эти режимы называются режимами с закливанием счета (импульсные генераторы).



Структурная схема программируемого таймера.

Программирование :

Программируемый таймер относится к классу функционально ориентированных программно управляемых интерфейсных БИС. Перед началом работы в него нужно загрузить управляющее слово и константу пересчета. Управляющее слово задает один из 6 режимов работы, тип счета, порядок загрузки и размерность константы пересчета (1 или 2 байта). Время-импульсная функция формируется на выходе OUT при GATA=1. Формирование функции производится с помощью счетчика таймера, работающего в режиме вычитания по срезу сигнала CLK. Программист может определить состояние каналов программируемого

таймера с помощью специальных команд :

- «защелка» (чтение на лету) - CLC;
- чтение состояния канала - RBC.

Эти команды позволяют, не прерывая счета, опросить состояние счетчика таймера СЕ. Команда RBC позволяет прочитать содержимое регистра состояния канала, разряды которого несут информацию о запрограммированном режиме, состоянии выхода OUT и флаге обновления.

Управляющее слово (CW) :

Управляющее слово загружается в регистры RCWn каналов программируемого таймера по командам вывода, формирующим на входах программируемого таймера коды :

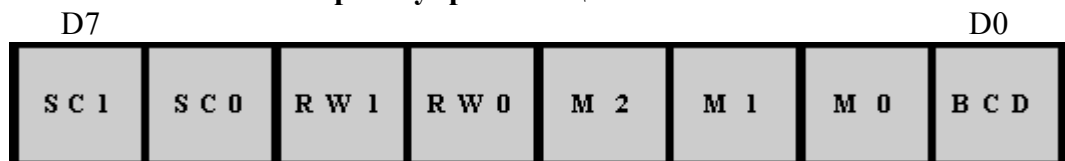
A0, A1

CS = 0

WR = 0

RD = 0

Формат управляющего слова



BCD - задает тип счета :

0 - двоичный

1 - двоично-десятичный

M2 M1 M0 - задают № режима :

0 0 0 - режим 0

0 0 1 - режим 1

x 1 0 - режим 2

x 1 1 - режим 3

1 0 0 - режим 4

1 0 1 - режим 5

RW1 RW0 - задают код порядка загрузки констант :

0 0 - код команды CLC

0 1 - чтение/запись старшего байта

1 0 - чтение/запись младшего байта

1 1 - чтение/запись младшего, затем старшего байтов

SC1 SC0 - задают № канала :

0 0 - канал 0

0 1 - канал 1

1 0 - канал 2

1 1 - код команды RBC

Управляющее слово загружается в тот канал программируемого таймера, адрес которого указывается в самом формате управляющего слова и сохраняется там все время работы или до следующего программирования.

Загрузка управляющего слова должна предшествовать загрузке констант.

В формате управляющего слова можно выделить 4 функциональных поля, с помощью которых задаются основные параметры работы канала:

- поле SC (D7,D6) - определяет адрес регистра RCWn конкретного канала; если код =11, то это команда для чтения состояния канала (RBC);
- поле RW (D5,D4) - определяет размерность и порядок загрузки констант; если в поле RW=00, то загружаемый код воспринимается как команда «защелка» (CLC);
- поле M (D3,D2,D1) - задает один из 6 режимов работы канала :
 - режим 0 - прерывание от таймера;
 - режим 1 - программируемый ждущий одновибратор;
 - режим 2 - импульсный генератор частоты;
 - режим 3 - генератор импульсов со скважностью 2;
 - режим 4 - программно запускаемый одновибратор;
 - режим 5 - аппаратно запускаемый одновибратор;
- поле BCD (D0) - определяет тип счета :
 - =0 - константа задается в двоичном коде и может принимать значения в диапазоне от 0 до 65535;
 - =1 - константа задается в двоично-десятичном коде и может принимать значения в диапазоне от 0 до 9999.

После загрузки управляющего слова необходимо загрузить в регистры каналов соответствующие константы пересчета. Порядок загрузки каналов управляющим словом и константами строго определен. Возможны 2 варианта :

- загрузка в любой последовательности управляющего слова, а затем констант пересчета;
- загрузка управляющего слова, загрузка констант и т.д.

Общие и обязательные требования для загрузки управляющего слова и констант:

- загрузка управляющего слова должна опережать загрузку констант;
- загрузка констант должна выполняться до конца, как определено разрядами RW.

Состояние счетчика таймера CEn можно прочесть тремя способами :

1) Чтение по обычным командам ввода позволяет прочесть состояние счетчика CEn в любое время. Выполняется с помощью обычных команд ввода с адресом, формирующим на входах A0,A1 код соответствующего канала. Необходимым условием для выполнения этой операции является остановка счета перед выполнением команды чтения (GATA=0). Операцию чтения необходимо выполнить до конца.

2) Чтение по CLC позволяет прочитать состояние CEn в любое время без остановки счета. Для этого нужно загрузить эту команду в определенный момент времени. Она загружается в программируемый таймер также, как и управляющее слово.

Формат управляющего слова (команда CLC).

D7				D0			
SC1	SC0	0	0	X	X	X	X

SC1 SC0 - задают № канала :

0	0	-	канал 0
0	1	-	канал 1
1	0	-	канал 2
1	1	-	код RBC

D7,D6 - задают адрес защелкиваемого канала. Они не могут принимать значения =11.

D5,D4 - задают код команды CLC.

D3,D2,D1,D0 - не участвуют в операции.

После загрузки CLC, выполняется операция чтения также, как и в 1-ом способе, но без снятия GATA. Прочитать CLn можно только после выполнения хотя бы одного цикла в CEn. Если операция чтения CLn не выполнена или выполнена не до конца, то новая команда CLC не будет восприниматься.

3) Чтение счетчика таймера СЕп по команде RBC - чтение состояния канала - позволяет в любой момент времени прочесть слово состояния канала SW и выполнить операцию «защелки» одного или нескольких каналов одновременно. Эта команда загружается в программируемый таймер так же, как и управляющее слово.

Формат управляющего слова (команда RBC).

D7							D0
1	1	COUNT	STAT	CNT2	CNT1	CNT0	O

D7,D6 - код команды RBC

D5 - если =0, то «защелка» канала, указанного в D1,D2,D3

D4 - если =0, то считать состояние каналов, указанных в D1,D2,D3

D3 - канал 2

D2 - канал 1

D1 - канал 0

Слово состояние канала (SW):

Каждый канал имеет регистр слова состояния RS, его формат:

D7							D0
O U T	F N	R W 1	R W 0	M 2	M 1	M 0	B C D

BCD - задает тип счета :

0 - ДВОИЧНЫЙ

1 - двоично-десятичный

M2 M1 M0 - задают № режима :

0 0 0 - режим 0

0 0 1 - режим 1

0 1 0 - режим 2

0 1 1 - режим 3

1 0 0 - режим 4

1 0 1 - режим 5

RW1 RW0 - задают код порядка загрузки констант :

0	0	- CLC
0	1	- чтение/запись старшего байта
1	0	- чтение/запись младшего байта
1	1	- чтение/запись младшего, затем старшего байтов
FN		- флаг перезагрузки констант канала :
0		- регистр CR свободен
1		- регистр CR занят
OUT		- состояние выхода канала

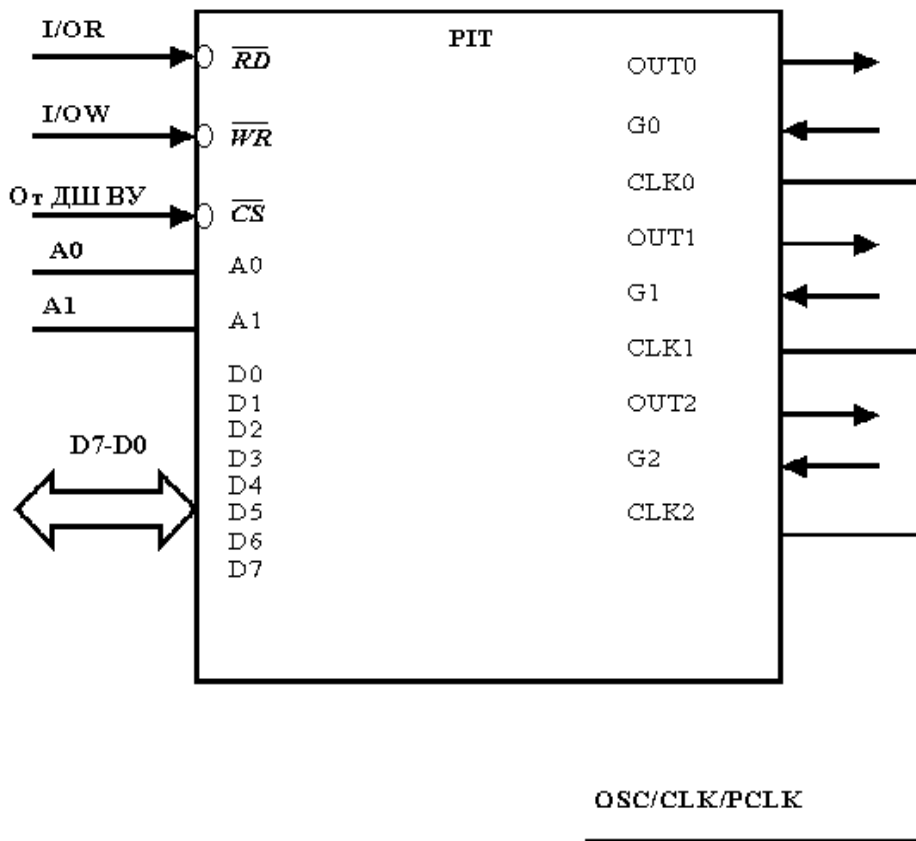
В процессе работы канала изменяются только 2 старших бита слова состояния (D7,D6). Остальные разряды соответствуют разрядам ранее загруженного управляющего слова, которое позволяет контролировать правильность его загрузки.

D7 - несет информацию о состоянии выхода OUT в момент выполнения команды RBC.

D6 - является флагом обновления регистра констант CRn. Этот флаг особенно необходим для режимов 1 и 5. Позволяет определить произошла ли загрузка констант из CRn в CEn или нет (загрузка производится по фронту GATA). Если FN=0, то произошла перезагрузка и CRn можно использовать для загрузки новой константы.

Результаты выполнения команды RBC читаются по адресам указанных в команде RBC каналов в следующем порядке: сначала RS, затем младший и старший байты счетчика.

Подключение программируемого таймер к микропроцессорной системе



СПРАВОЧНЫЕ МАТЕРИАЛЫ ПО ПОДСИСТЕМЕ ПРОГРАММИРУЕМОГО ТАЙМЕРА IBM PC/AT

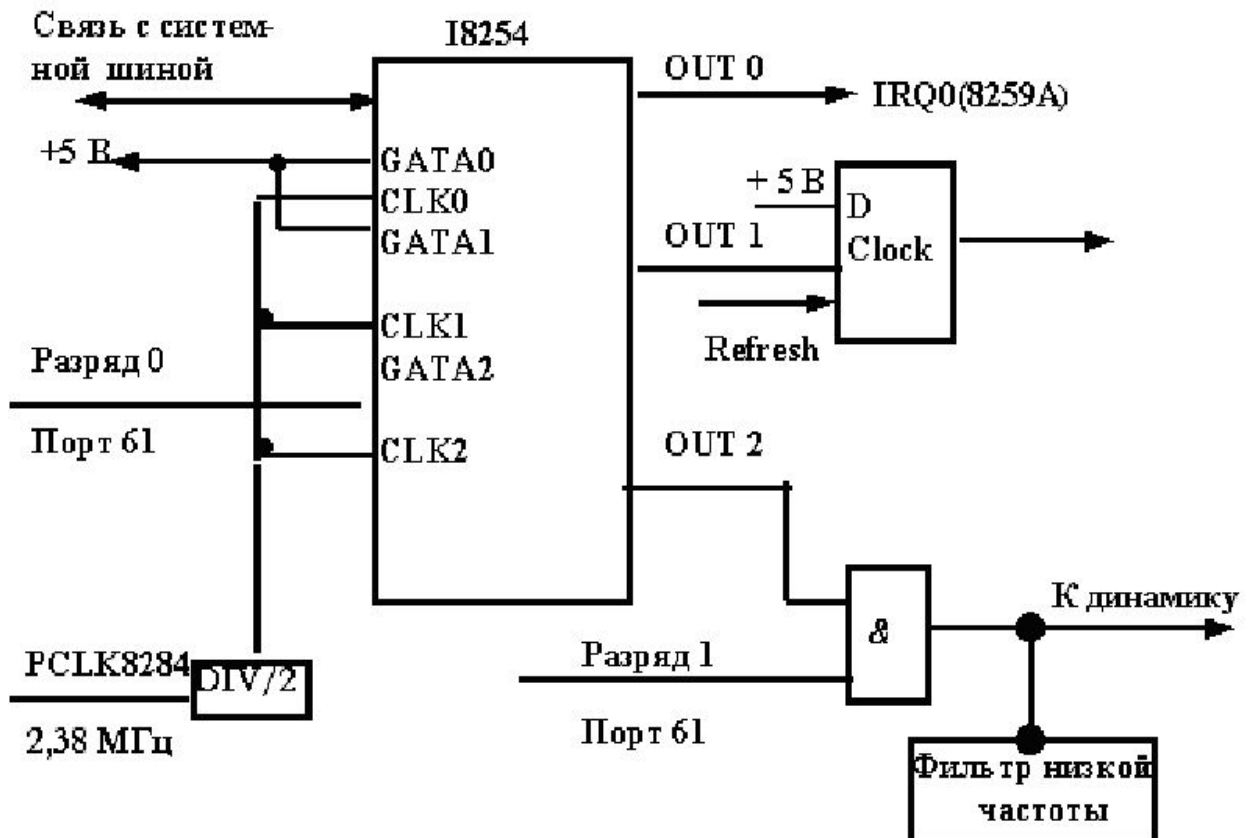
Общие характеристики :

Архитектура PC/AT использует трехканальный таймер i8254 в качестве системного таймера. В PC/AT входные синхросигналы для всех трех каналов поступают от единого источника с частотой $f=1,19$ МГц. Каждый канал может программироваться для работы в одном из 6 режимов.

В PC/AT выход канала 0 связан с IRQ 0 контроллера прерываний 8259A и обеспечивает поддержку системного времени, при этом используется режим 3 таймера - генератор меандра.

Выход канала 1 генерирует сигнал запроса регенерации динамической памяти, при этом используется режим 2 таймера - импульсный генератор частоты.

Выход канала 2 генерирует тоновый сигнал для динамика, при этом используется режим 3 таймера - генератор меандра.



Порты, используемые для программирования таймера:

Адрес порта	Функция
040 h	Чтение/Запись содержимого счетчика канала 0
041 h	Чтение/Запись содержимого счетчика канала 1
042 h	Чтение/Запись содержимого счетчика канала 2
043 h	Запись управляющего слова и команд CLC и RBC

2.8.2. Примеры программ:

Примечание: В программах специально допущены ошибки (логические), которые студентам необходимо исправить в процессе выполнения работы.

Программа 1.

(Периодическое изменение тональности звукового сигнала в режиме работы 3)

	mov	DX,100	- Внешний цикл
	mov	AL,B6	
	out	[43],AL	- Программируем таймер
	mov	AL,4B	
	out	[61],AL	- Разрешаем звук
next:	mov	AL,10	- Содержимое AL=10
	out	[42],AL	- Занесение по адресу 42h константы 1010h (мл. байта)
	out	[42],AL	- Занесение по адресу 42h константы 1010h (ст. байта)
	mov	CX,20	- Длительность звучания (f=1,19МГц/ 1010h)
a:	loop	a	- Задержка
	mov	AL,25	- Содержимое AL=25
	out	[42],AL	- Занесение по адресу 42h константы 2525h (мл. байта)
	out	[42],AL	- Занесение по адресу 42h константы 2525h (ст. байта)
	mov	CX,20	- Длительность звучания (f=1,19МГц/2525h)
b:	loop	b	- Задержка
	dec	DX	- Уменьшение DX
	jnz	next	- Если не равно 0, то переход на next:
	mov	AL,49	- Содержимое AL=49
	out	[61],AL	- Выключение звука
	int	3	

Программа 2.

(Периодическое выдачи сигналов оразличной длительности в режиме работы 0)

mov	DX,100	- Внешний цикл
mov	AL,C0	
out	[43],AL	- Программируем таймер
mov	AL,4B	

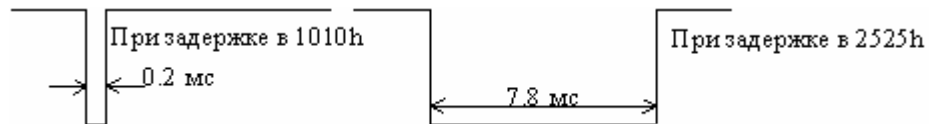
	out	[61],AL	- Программируем таймер
next:	mov	AL,01	- Занесение константы =0101h по адресу 42h
	out	[42],AL	
	out	[42],AL	
	mov	CX,100	
a:	loop	a	- Задержка
	mov	AL,25	- Занесение константы =2525h по адресу 42h
	out	[42],AL	
	out	[42],AL	
	mov	CX,100	
b:	loop	b	- Задержка
	dec	DX	- Уменьшение DX
	jnz	next	- Если не равно 0, то переход на next:
	mov	AL,49	- Выключение звука
	out	[61],AL	
	int	3	

Программа 3.

(РЕЖИМ 0 - прерывание терминального счета)

	MOV	DX,FFFF	-внешний цикл
	MOV	AL,B1	-
	OUT	[43],AL	-запись управляющего слова
NEXT:	MOV	AL,10	- константа счета
	OUT	[42],AL	- младший байт константы
	OUT	[42],AL	- старший байт константы
	MOV	AL,4B	-
	OUT	[61],AL	-разрешение звука
	MOV	CX, F	
A:	LOOP	A	- задержка
	MOV	AL,25	- константа счета
	OUT	[42],AL	- младший байт константы
	OUT	[42],AL	- старший байт константы
	MOV	CX,FF	-
B:	LOOP	B	- задержка
	DEC	DX	- внешний цикл
	JNZ	NEXT	- проверка на ноль
	MOV	AL,49	-
	OUT	[61],AL	- запрет звука
	INT	3	- завершение

Осциллограммы:

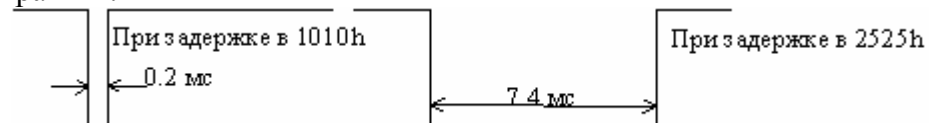


Программа 4.

(РЕЖИМ 1 - программируемый ждущий мультивибратор)

	MOV	DX,F	- внешний цикл
	MOV	AL,B3	-
	OUT	[43],AL	- запись управляющего слова
NEXT:	MOV	AL,01	- константа счета
	OUT	[42],AL	- младший байт константы
	OUT	[42],AL	- старший байт константы
	MOV	AL,4B	-
	OUT	[61],AL	- разрешение звука
A:	MOV	CX,F	-
	LOOP	A	- задержка
	MOV	AL,4A	-
	OUT	[61],AL	- запрет звука
	MOV	AL,25	- константа счета
	OUT	[42],AL	- младший байт константы
	OUT	[42],AL	- старший байт константы
	MOV	AL,4B	-
	OUT	[61],AL	- разрешение звука
B:	MOV	CX,FF	-
	LOOP	B	- задержка
	MOV	AL,4A	-
	OUT	[61],AL	- запрет звука
	DEC	DX	- внешний цикл
	JNZ	NEXT	- проверка на ноль
	INT	3	- завершение

Осциллограммы:



Программа 5.

(РЕЖИМ 2 - импульсный генератор частоты)

Команды AFD:

O 43,B5 - запись управляющего слова

O 42,10 - младший байт константы

O 42,10 - старший байт константы

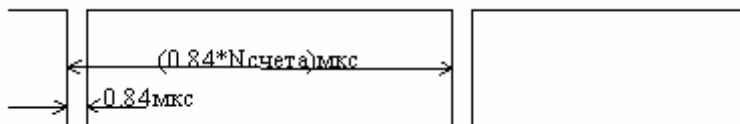
O 61,4B - разрешение выхода OUT2 и выхода на динамик

MOV	AL,B5	-
OUT	[43],AL	- запись управляющего слова
MOV	AL,10	- константа счета
OUT	[42],AL	- младший байт константы
OUT	[42],AL	- старший байт константы
MOV	AL,4B	-
OUT	61,4B	- разрешение OUT2 и выхода на динамик
INT 3		- завершение

Осциллограмма:

Из-за влияния паразитных сопротивлений и длинной линии (соединительные провода) на экране осциллографа отрицательный прямоугольный импульс длительностью 0.84 мкс не виден:

В идеале:



Программа 6.

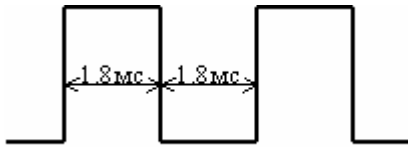
(РЕЖИМ 3 - генератор меандра)

Команды AFD:

O 43,B7 - запись управляющего слова
 O 42,10 - младший байт константы
 O 42,10 - старший байт константы
 O 61,4B - разрешение выхода OUT2 и выхода на динамик

MOV	AL,B7	-
OUT	[43],AL	- запись управляющего слова
MOV	AL,10	- константа счета
OUT	[42],AL	- младший байт константы
OUT	[42],AL	- старший байт константы
MOV	AL,4B	-
OUT	[61],4B	- разрешение OUT2 и выхода на динамик
INT 3		- завершение

Осциллограмма:



$T=3.6 \text{ мс}$

$f=280 \text{ Гц}$

$f_{clk}=1.19 \text{ МГц}$

$1010h=4112$

$f=f_{clk}/N=289.5 \text{ Гц}$

Программа 7.

(РЕЖИМ 4 - программно-формируемый строб)

	MOV	DX,1000	- внешний цикл
	MOV	AL,B9	-
	OUT	[43],AL	- управляющее слово
	MOV	AL,10	- константа счета
NEXT:	OUT	[42],AL	- младший байт константы
	OUT	[42],AL	- старший байт константы
	MOV	AL,4B	-
	OUT	[61],AL	- разрешение OUT
	MOV	CX,F	-
A:	LOOP	A	- задержка
	DEC	DX	- внешний цикл
	JNZ	NEXT	- проверка на ноль
	INT	3	- завершение

Осциллограмма:

Из-за влияния паразитных сопротивлений и длинной линии (соединительные провода) на экране осциллографа отрицательный прямоугольный импульс длительностью 0.84 мкс не виден:



в идеале:

Программа 8.

(РЕЖИМ 5 - аппаратно-формируемый строб)

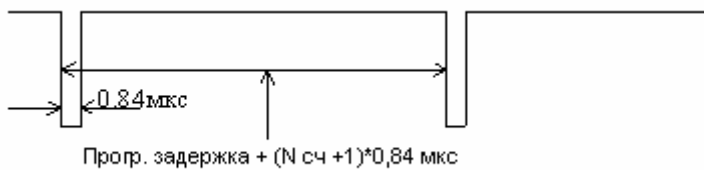
MOV	DX,FFFF	- внешний цикл
MOV	AL,BB	-
OUT	[43],AL	- управляющее слово

NEXT:	MOV	AL,[10]	- константа счета
	OUT	[42],AL	- младший байт константы
	OUT	[42],AL	- старший байт константы
	MOV	AL,3B	-
	OUT	[61],AL	- разрешение OUT
	MOV	CX,F	
A:	LOOP	A	- задержка
	MOV	AL,4A	-
	OUT	[61],AL	- запрет OUT
	MOV	AL,4B	-
	OUT	[61],AL	- разрешение OUT
	DEC	DX	- внешний цикл
	JNZ	NEXT	- проверка на нуль.
	INT 3		- завершение

Осциллограмма:

Из-за влияния паразитных сопротивлений и длинной линии (соединительные провода) на экране осциллографа отрицательный прямоугольный импульс длительностью 0.84 мкс не виден:

В идеале:

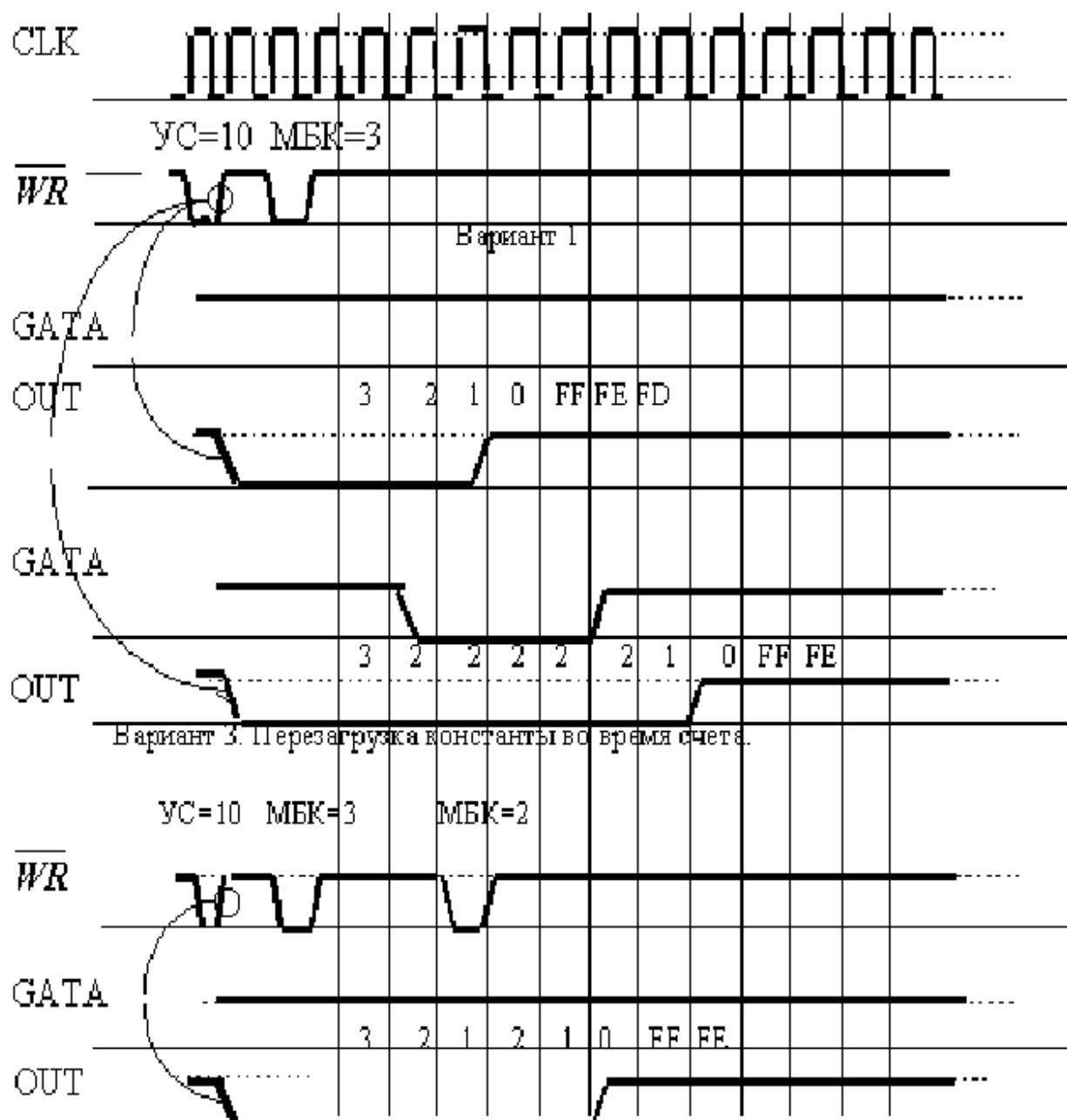


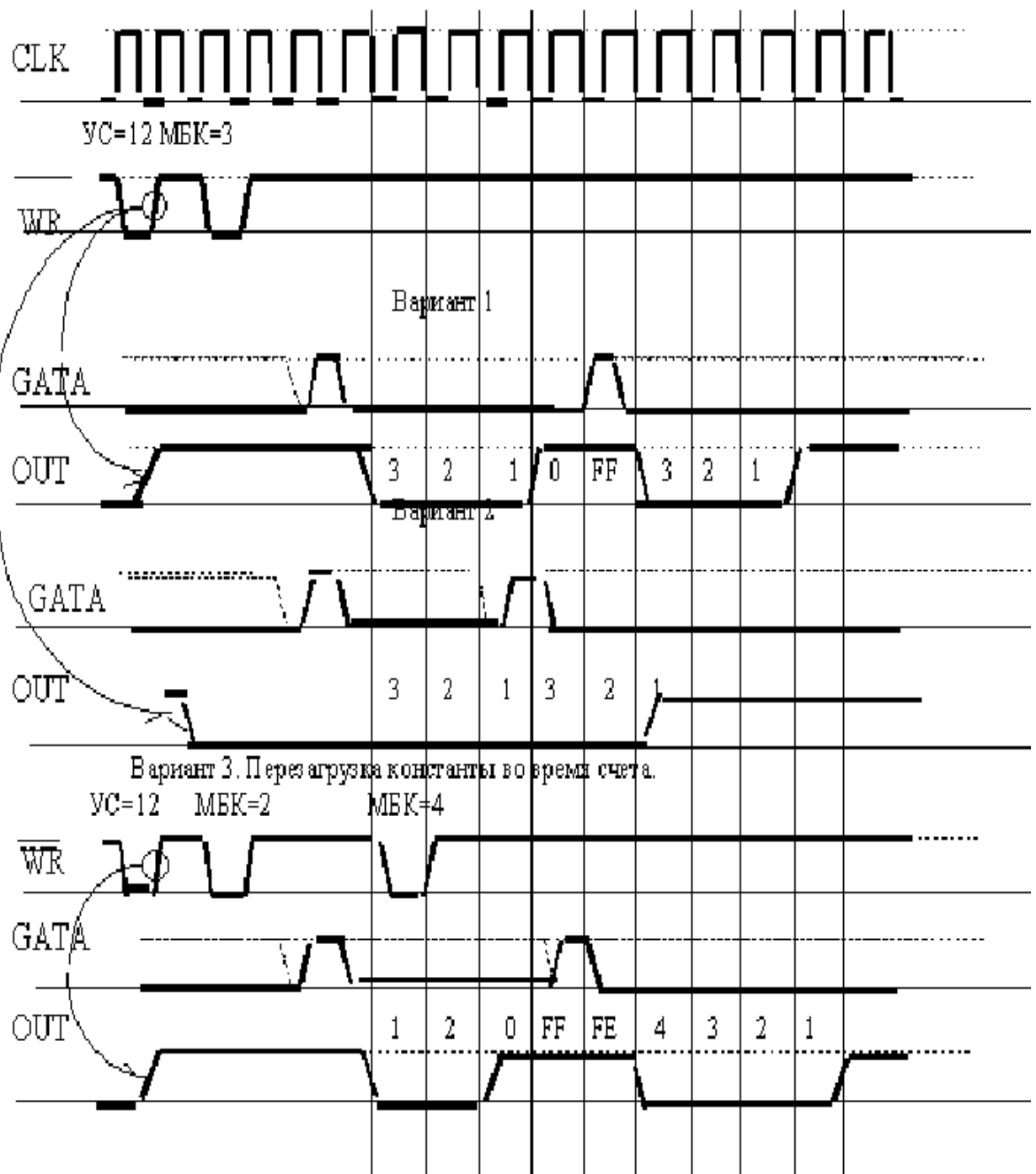
2.8.3. Примеры временных диаграмм работы программируемого таймера в разных режимах:

Режим 0:

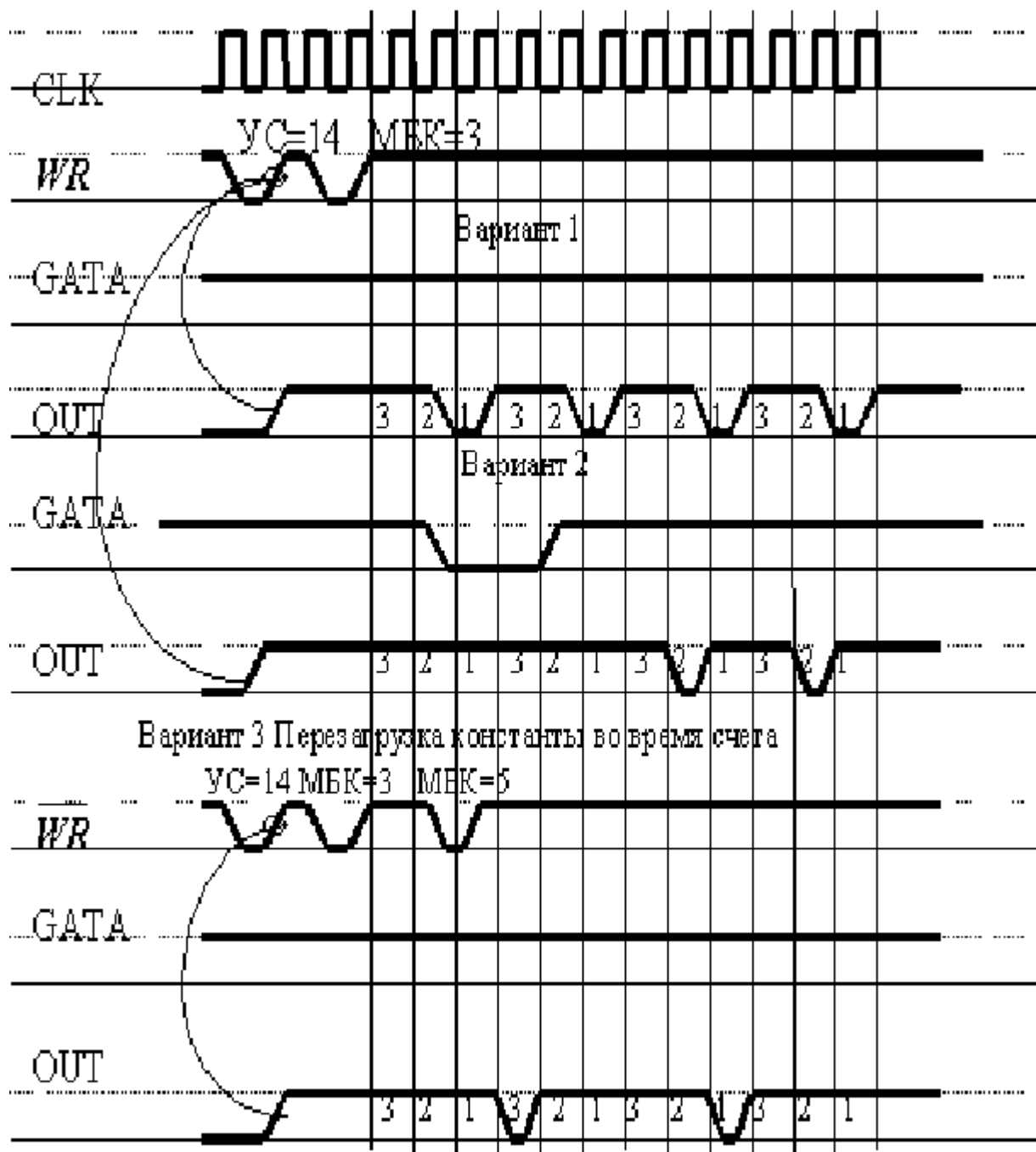
УС-управляющее слово

МБК-младший байт константы



Режим 1:

Режим 2:



Резюме 3:

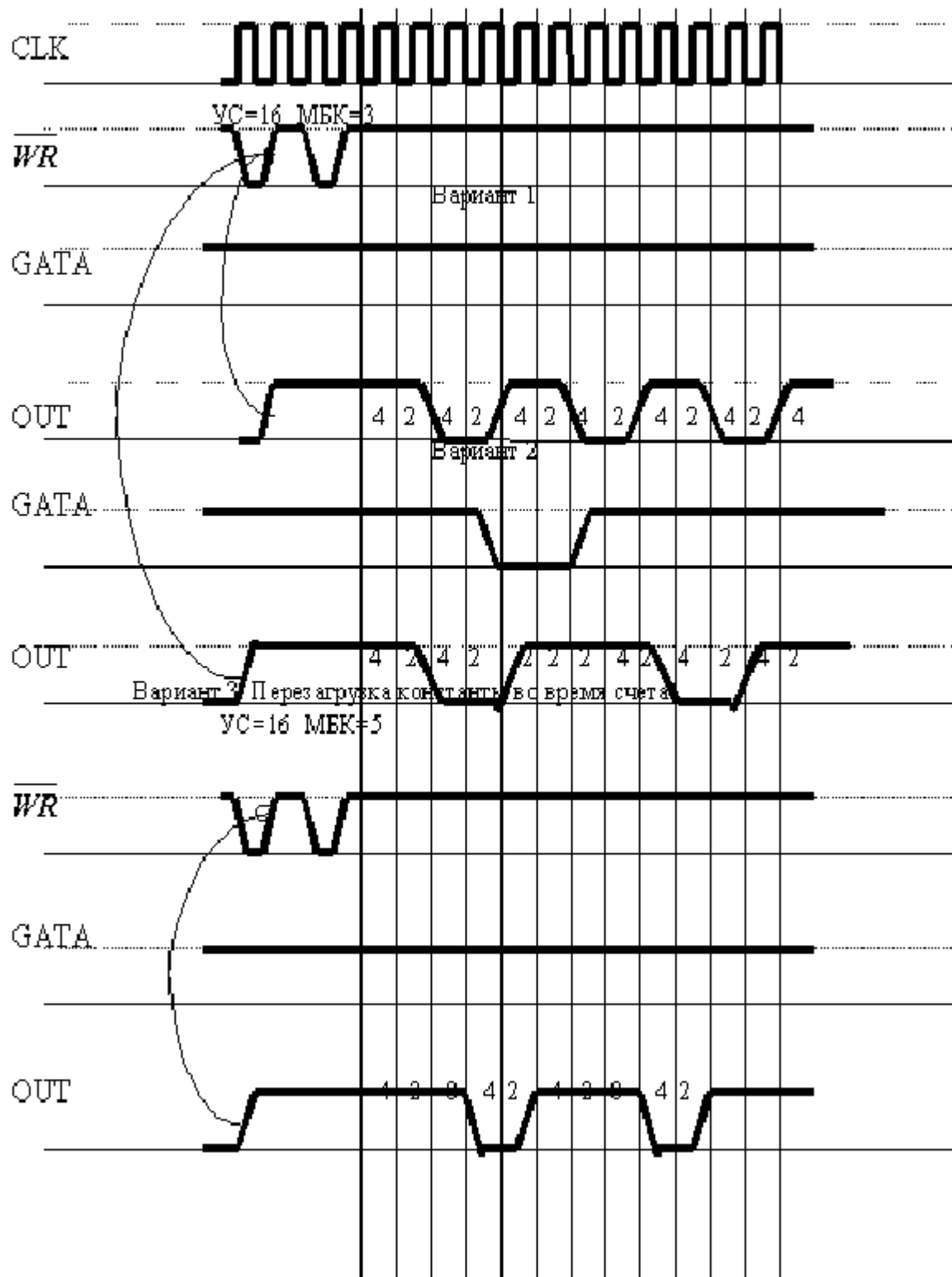
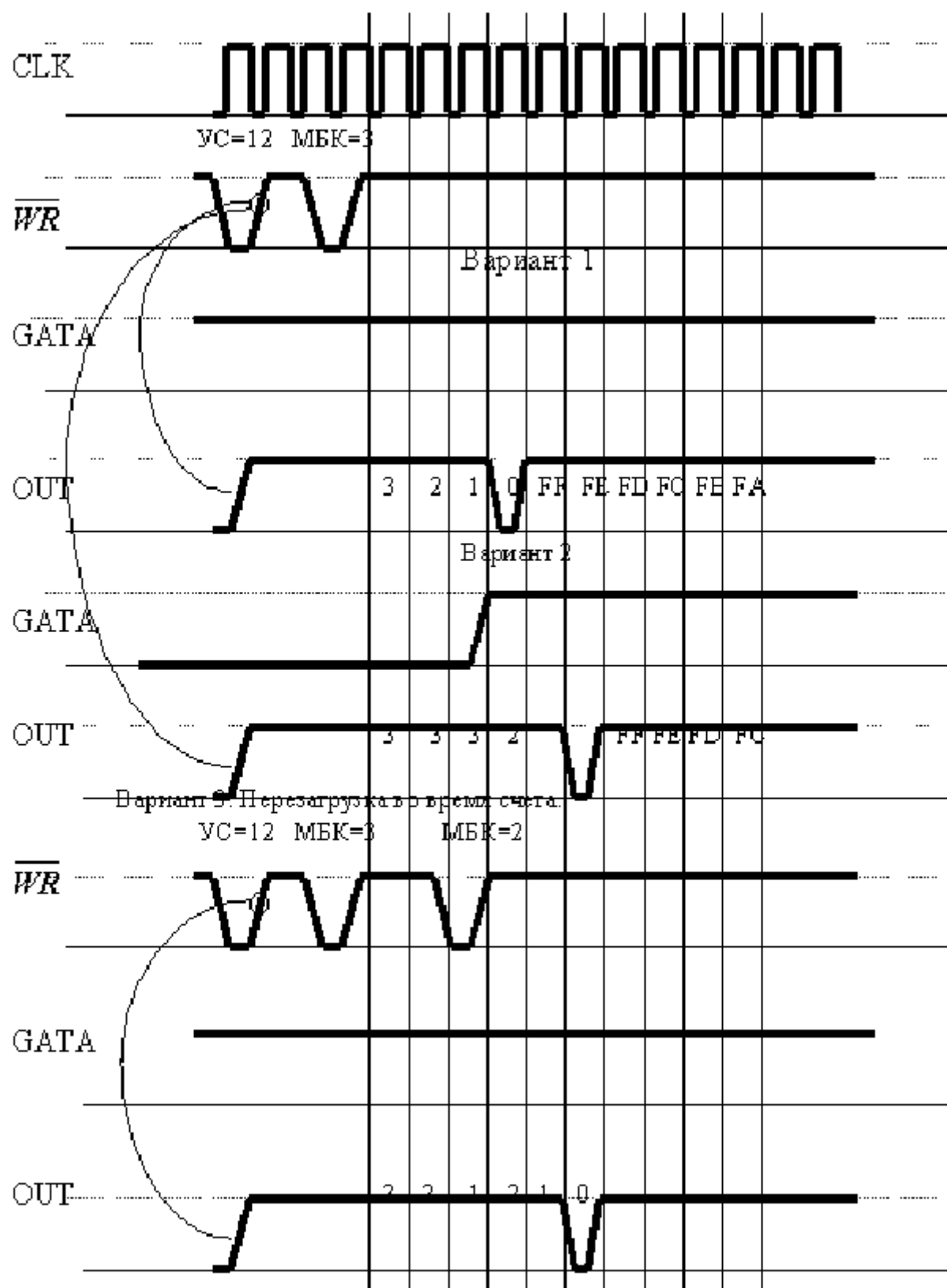
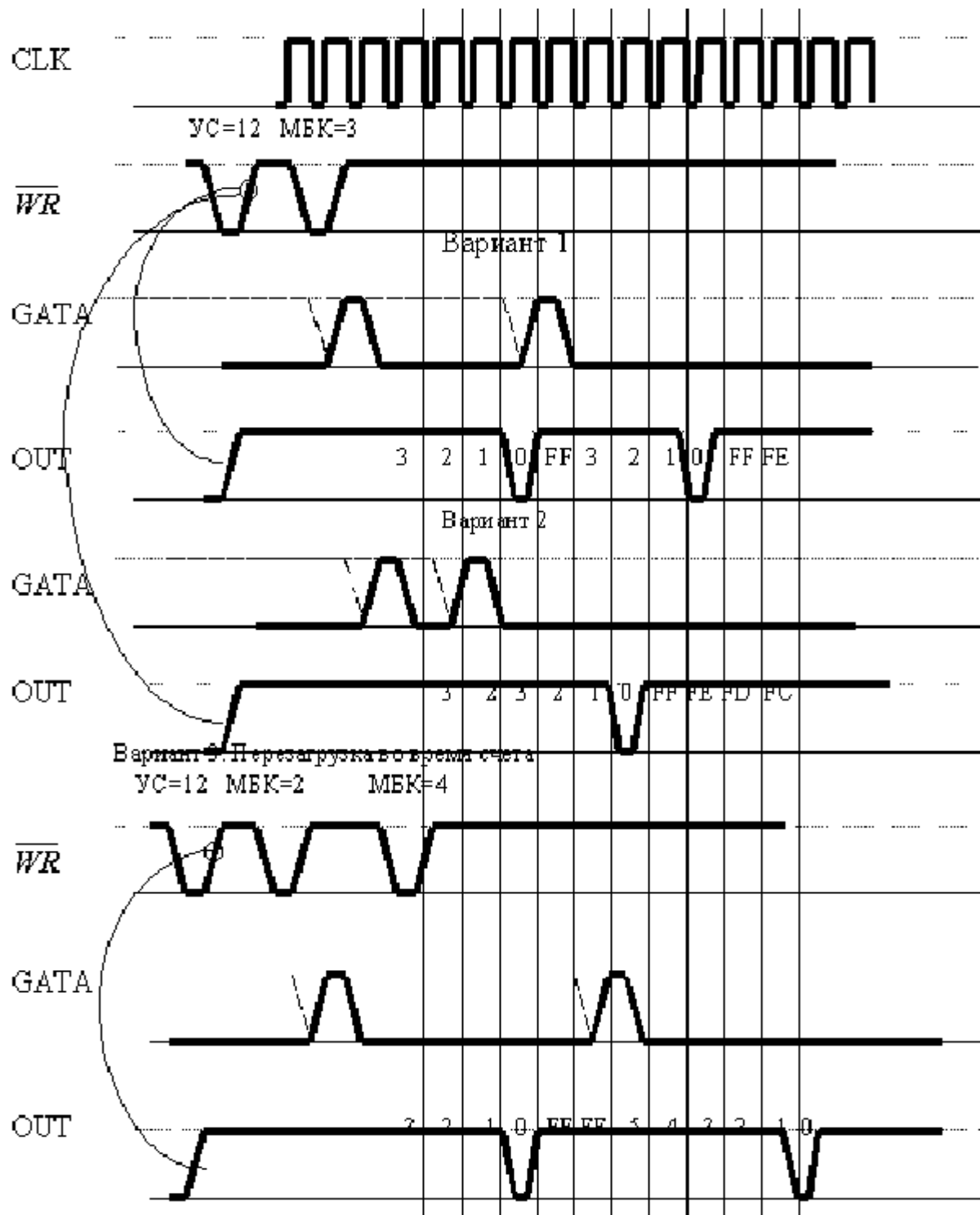


Рисунок 4:

Режим 5:

Лабораторная работа 2 (описание)

3.«Универсальный синхронно-асинхронный
приемопередатчик (УСАПП) КР580ВВ51 в составе
адаптера АПД ПЭВМ
«Искра-1031»».

3.1. ЦЕЛЬ РАБОТЫ.

Целью настоящей работы является ознакомление с принципами работы универсального синхронно-асинхронного приемопередатчика (УСАПП) КР580ВВ51А, входящего в состав адаптера ААПД ПЭВМ «Искра 1031», а также с приемами его программной настройки на различные режимы работы (совместно с программируемым таймером КР580ВИ53 (КР1810ВИ54)).

3.2. ТЕХНИЧЕСКИЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ В РАБОТЕ.

В лабораторной работе используется ПЭВМ «Искра 1031», ААПД 3.093.142, предназначенный для использования в составе ПЭВМ «ИСКРА 1031», осциллограф С1-81, устройство коммутирующее 5.284.011.

3.3. ОСНОВНЫЕ ЭЛЕКТРИЧЕСКИЕ ПАРАМЕТРЫ БИС КР580ВВ51.

- | | |
|---|-------------|
| 1)Выходное напряжение логического нуля UOL | < 0.4 |
| 2)Выходное напряжение логической единицы UOH ,В | > 2.4 |
| 3)Ток потребления от источника питания ICC ,mA | < 60 |
| 4)Ток утечки на управляющих входах IIL ,mkA | - 10 ... 10 |

3.4. НАЗНАЧЕНИЕ УСАПП КР580ВВ51 и его использование в СОСТАВЕ ААПД.

Универсальный синхронно-асинхронный приемопередатчик (УСАПП) предназначен для организации обмена между микропроцессором и внешним устройством в последовательном формате.

УСАПП может принимать данные с 8-ми разрядной шины данных микропроцессора и передавать их в последовательном формате внешнему устройству, а также получать последовательные данные от внешнего устройства и передавать их в параллельном формате в микропроцессор.

Обмен данными в асинхронном режиме идет с максимальной скоростью 9600 бит/сек, а в синхронном - 56000 бит/сек.

Длина передаваемых символов от 5 до 8 бит. При передаче данных в микропроцессор символов меньше 8 бит не использующиеся биты заполняются нулями.

В асинхронном режиме формат включает служебные биты: стартовый, бит контроля четности и стоповые биты.

В составе ААПД используется семь микросхем (БИС) УСАПП (см. Структурную схему ААПД и Схему электрическую принципиальную АПД 3.093.142 ЭЗ ТО ААПД).

На электрической принципиальной схеме АПД 3.093.142 ЭЗ УСАПП обозначены как IOS 0 ...IOS 6. На их базе организованы стандартные последовательные интерфейсы С2 (RS-232C) и ИРПС (Current Loop):

- ☐ узел приема - передачи С2 на микросхеме D20 (IOS 0);
- ☐ узел приема - передачи
 - ИРПС 1 - D21 (IOS 1);
 - ИРПС 2 - D22 (IOS 2);
 - ИРПС 3 - D23 (IOS 3);
 - ИРПС 4 - D24 (IOS 4);

ИРПС 5 - D25 (IOS 5);

ИРПС 6 - D26 (IOS 6);

БИС D20... D26 работают совместно с БИС программируемого таймера КР580ВИ53, который формирует для УСАПП тактовые импульсы приема-передачи данных, задавая тем самым стандартные скорости обмена согласно табл. 7 ТО ААПД.

На электрической принципиальной схеме он обозначен как SYN1 (D17). Формируемые им совместно с мультиплексором D31 (K555КП2) последовательности импульсов f1, f2, f3, f4, f5 подаются, соответственно, на входы синхронизации приема и передачи УСАПП (IOS 0 ... IOS 6):

f1 - на D31, которая формирует сигналы f4 и f5,

f2 - на входы ТЕС и RCC микросхем D21 (IOS 1), D22 (IOS 2), D23 (IOS 3),

f3 - на входы ТЕС и RCC микросхем D24 (IOS 4) и D25 (IOS 5),

f4 - на входы ТЕС микросхем D20 (IOS 0) и D26 (IOS 6),

f5 - на входы RCC микросхем D20 (IOS 0) и D26 (IOS 6).

Описание принципов работы и программирования БИС КР580ВИ53 приведено в описании лабораторной работы: «БИС программируемого таймера КР580ВИ54 (I8254)».

Для реализации интерфейса C2 (RS-232C) совместно с УСАПП (D20) используется БИС параллельного программируемого порта КР580ВВ55, которая обеспечивает выдачу и прием управляющих и вспомогательных сигналов интерфейса C2 (RS-232C). На электрической принципиальной схеме она обозначена как IOP (D19).

Описание принципов работы и программирования БИС КР580ВВ55 приведено в описании лабораторной работы «БИС параллельного программируемого интерфейса КР580ВВ55 в составе ААПД».

Режимы работы последовательных интерфейсов по запросам на прерывание обеспечиваются подключением выводов ERD УСАПП (D20...D26) ко входам запросов на прерывание БИС контроллера прерывания КР1810ВН59А, обозначенной на электрической принципиальной схеме как INR (D27). Для обеспечения приема и передачи двуполярных импульсов, предусмотренных стандартом, в схеме интерфейса C2 (RS-232C) используются микросхемы преобразователей уровней К170УП2 (D38...D40) и К170АП2 (D32...D36).

Для обеспечения требований стандарта ИРПС на прием и передачу токовых импульсов с гальванической развязкой цепей приема и передачи используются транзисторы типа КТ3102Б, КТ3107Б и оптроны типа АОД129А (см. электрическую принципиальную схему ААПД).

Общая синхронизация работы БИС IOS 0... IOS6 (D20...D26) и SYN (D17,D18) осуществляется сигналом F1 с частотой 1228,8 кГц, формируемым генератором тактовых импульсов (D15)(2457,8 кГц) и делителем (D16.1).

Входные и выходные сигналы интерфейсов C2(RS-232C) и ИРПС1...ИРПС6 выведены на внешний разъем X3 УСАПП.

Распределение сигналов по контактам разъема X3 для сигналов стандарта C2 (RS-232C) приведены в таблице 1 и 2, а для стандарта ИРПС (Current Loop)-в таблице 3 ТО ААПД.

Для практической работы к разъему X3 подключается коммутирующее устройство 5.284.011. (коннектор).

Разводка сигналов по контактам разъемов коннектора (2 разъема C2) показана в табл. 1.

Рекомендуемое соединение адаптера АПД и коннектора приведено на рис. 1.

Таблица 1

Контакт	Цепь	Выводы ППИ
1	Цепь 101 (201)	
2	Цепь 103 (203)	РА6, ВХ
3	Цепь 104 (204)	РА5, ВХ
4	Цепь 105 (205)	РА4, ВХ
5	Цепь 106 (206)	РВ5, ВЫХ
6	Цепь 107 (207)	РВ4, ВЫХ
7	Цепь 102 (202)	РВ7, ВЫХ
8	Цепь 109 (209)	РВ2, ВЫХ
9	Цепь 108 (208)	РВ3, ВЫХ
10	Цепь 114 (210)	РА3, ВХ
11	Цепь 126	РВ1, ВЫХ
12	Цепь 115 (211)	РВ6, ВЫХ
13	Цепь 120 (213)	РА7, ВХ
14	Цепь 125	РА2, ВХ

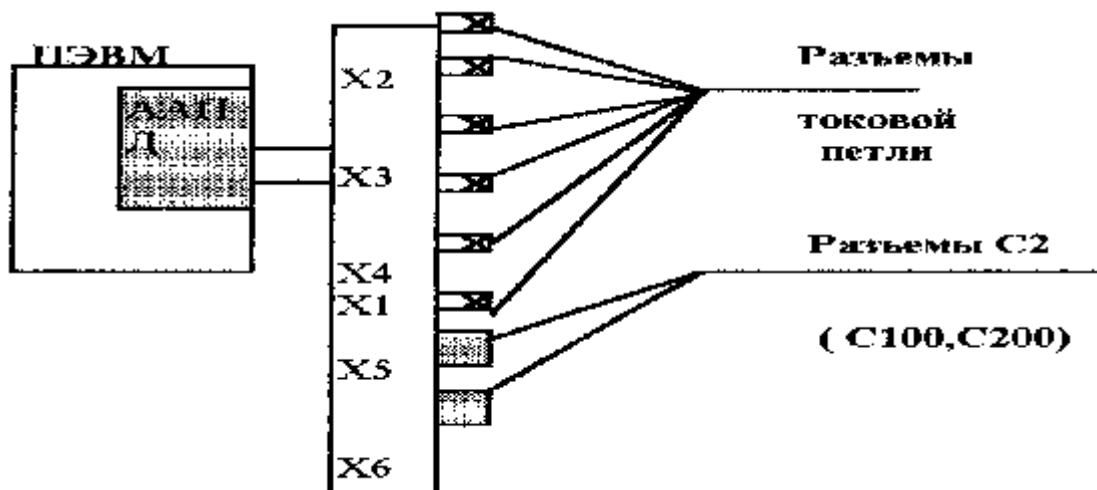


Рис. 1. Схема подключения коннектора.

Программирование УСАПП (IOS 0...IOS6), таймера (SYN1) и ППИ (IOP).

Программирование УСАПП, таймера и ППИ осуществляется записью в управляющие регистры этих устройств управляющей информации.

Обмен информацией между процессором и внешним устройством осуществляется через соответствующие информационные регистры (регистры данных) УСАПП и ППИ.

Информация о состоянии БИС получается путем считывания содержимого соответствующих регистров состояния.

Адреса регистров IOS 0 ...IOS 6, SIN 1, SIN 2, IOP и INR приведены в таблице 6 ТО ААПД.

3.5. УКАЗАНИЕ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ.

Лабораторная работа проводится на ПЭВМ «Искра 1031», в которой установлена MS DOS версии 3 и выше. Дополнительно устанавливается программа - отладчик AFD. К установленной в системный блок плате адаптера АПД должен быть подключен коннектор. Контроль выходных сигналов интерфейса ИРПС осуществляется с токосъемного сопротивления, подключенного к разъему канала 5.

Этот контроль осуществляется с помощью электронного осциллографа С1 - 81.

3.6. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ.

3.6.1 Изучить техническое описание и схему электрически принципиальную адаптера АПД.

3.6.2 По техническому описанию и схеме электрически принципиальной определить адреса канала таймера SYN1, синхронизирующего работу УСАПП канала 5 адаптера АПД по входам TEC(9) и RCC(25), микросхема IOS5 (УСАПП канала 5) и соответствующие адреса регистров таймера и УСАПП.

3.6.3 Определить константы, обеспечивающие работу таймера SYN1 при двоично-десятичном счете и коэффициенте деления в УСАПП 1:1 по синхронизации УСАПП на скорости передачи и приема данных 9600 бит/сек.

3.6.4 Определить управляющие слова и команды, которые необходимо подать на УСАПП канала 5, чтобы настроить его на работу по передаче данных в асинхронном режиме со скоростью 9600 бит/с при наличии одного, полутора, двух стоп-битов ; длине слова 5, 6, 7 или 8 бит ; наличии и отсутствии контроля по четности.

3.6.5 Составить на ассемблере программу инициализации таймера SYN1.

3.6.6 Составить программу инициализации УСАПП канала 5 на выбранный режим работы.

Примечание : программа должна иметь блок программного сброса УСАПП в исходное состояние, блок задания инструкции режима (управляющего слова), блок подачи командного слова и блок вывода информации через УСАПП в асинхронном режиме в последовательном коде. При этом использовать константы : AA, 55, FF, 00 .

3.6.7 Отладить программу в среде AFD.

3.6.8 Провести исследование сигнала на выходе УСАПП по осциллографу в различных режимах работы УСАПП и различных выводимых символах (AA, 55, FF, 00). Осциллограммы на выходе УСАПП зарисовать. Оценить их реальные параметры и сравнить с расчетными (ожидаемыми).

Примечание : Для исследования сигналов по осциллографу, часть программы, отвечающую за вывод символа через УСАПП, должна быть зациклена (Выход из цикла по Ctrl-Esc в среде AFD).

3.6.9 Отлаженный вариант программы сохранить на диске под именем <filename.lbr> в двоичном виде, а также вывести на печать ее дизассемблированный вариант (пользоваться только средствами отладчика AFD).

3.6.10 Аналогичное задание выполнить для других каналов. Скорость передачи данных и номер канала задает преподаватель.

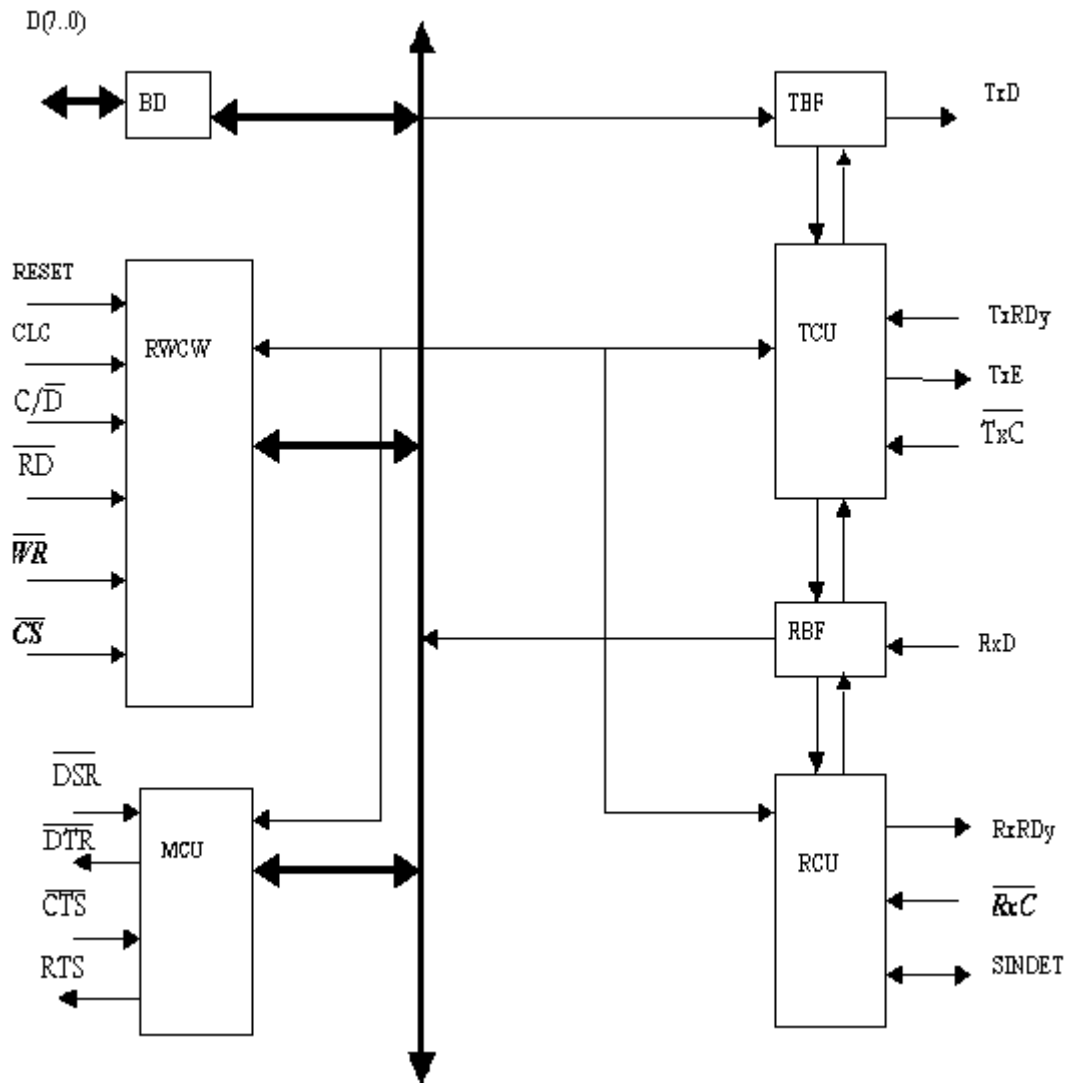
3.6.11 Оформить отчет по работе.

3.7. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.

- 1) Справочник «Микропроцессоры и микроЭВМ в системах автоматического управления» под ред. С.Т. Хвоща. Машиностроение, 1987г.
- 2) Лекции по курсу «Адаптеры и контроллеры ЭВМ»
- 3) Адаптер АПД. Техническое описание 3.093.138 ТО

3.8.ПРИЛОЖЕНИЕ

3.8.1 Структурная схема УСААП.



В состав УСААП входят:

1. Буфер передатчика TBF со схемой управления передатчиком TCU для приема данных от микропроцессора в параллельном формате и выдача их в последовательном формате на выход TXD.
2. Буфер приемника RBF со схемой управления RCU, выполняющий прием последовательных данных со входа RXD и передачу в микропроцессор в параллельном формате.
3. Буфер данных BD, представляющий собой параллельный 8-ми разрядный регистр с тремя состояниями, служащий для обмена данными и управляющей информацией между

микропроцессором и УСАПП.

4. Блок управления записью/чтением RWCU, принимающий управляющие сигналы от микропроцессора и генерирующий внутренние сигналы управления.

5. Блок управления модемом MCU, обрабатывающий управляющие сигналы, предназначенные для внешнего устройства.

Сигналы, используемые в данной схеме:

D(7..0) - канал данных

RESET (SR) - установка в исходное состояние или сброс.

CLOCK (SYN) - синхронизация со стороны системной магистрали.

C/D (A) - управление /данные.

«низкий уровень» - запись/чтение данных

«высокий уровень» - запись/чтение слов управления и состояния

RD - чтение из УСАПП в микропроцессор

WR - запись из микропроцессора в УСАПП

CS (CR) - выбор микросхемы (низким уровнем УСАПП подключается к системной магистрали).

DSR - готовность передатчика терминала

DTR - запрос передатчика терминала

CTS - готовность приемника терминала

RTS - запрос приемника терминала

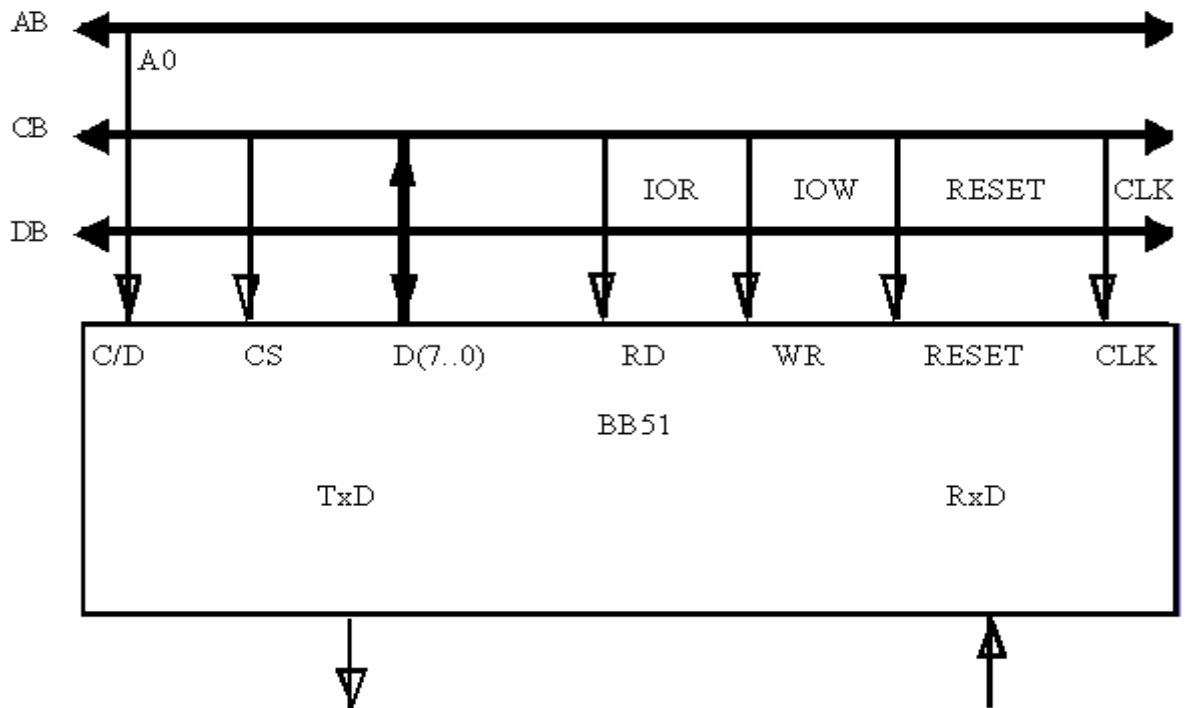
SINDET - вид синхронизации (для синхронного режима выходное напряжение высокого уровня - признак внутренней синхронизации, для синхронного режима с внешней синхронизацией сигнал является входным, в асинхронном - сигнал только выходной).

RXC (RCC) - синхронизация приемника (поступает с таймера)

RXRDY (ERD) - готовность приемника

RXD (RCD) - вход приемника

Подключение УСАПП к системной магистрали.



TXC (TEC) - синхронизация передатчика (поступает с таймера)
 TXE - конец передачи (высокий уровень на выходе признак окончания посылки данных)
 TXD (TED) - выход передатчика.
 TXRDY - готовность передачи.

3.8.2 Функционирование KP580BB51. Основные сигналы и управляющие слова.

Основные сигналы управления работой УСАПП подаются на блок RWCU от микропроцессора и определяют вид обрабатываемой информации и направление передач.

Операция	Сигналы			
	C/D	RD	WR	CR
Чтение из УСАПП в микропроцессор	0	0	1	0
Запись данных из МП в УСАПП	0	1	0	0
Чтение слова-состояния изУСАПП в МП	1	0	1	0
Запись управляющего слова из МП в УСАПП	1	1	0	0
Подключение	X	1	1	0
Отключение	X	X	X	1

Режимы работы УСАПП задаются программно, путем загрузки управляющих слов из микропроцессора.

Управляющие слова бывают двух видов:

1. Инструкция режима.
2. Команды.

Инструкция режима задает:

- а) синхронный или асинхронный режим работы
- б) формат данных
- в) скорость передачи/приема
- г) необходимость контроля

Инструкция заносится сразу после установки УСАПП в исходное состояние программно или по сигналу RESET и заменяется лишь при смене режима. Команда осуществляет управление установленным режимом обмена и может многократно задаваться в процессе обмена, управляя различными его этапами.

При асинхронном обмене команда загружается сразу же после инструкции режима, а при синхронном, перед ней располагается 1 или 2 синхросимвола. Ограничения на последовательность загрузки управляющих слов связано с внутренней организацией УСАПП.

В асинхронном режиме формат данных включает стоповые биты, биты данных от 5 до 8, контрольный бит и стоп биты.

Число битов данных и стоп битов, а также наличие или отсутствие контрольного бита задается инструкцией режима.

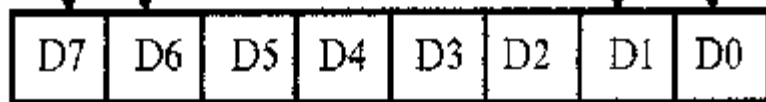
Формат инструкции режима для асинхронного обмена приведен на рис. 1.

Количество стоп битов:

Запрещ.	1	1.5	2
0	0	1	1
0	1	0	1

Деление CLK:

1/1	1/16	1/64
0	1	1
1	0	1



Контроль по четности:

по четн.	по нечетн	нет
1	0	x
1	1	0

Кол-во бит передачи

5	6	7	8
0	1	0	1
0	0	1	1

Рис. 1.

Разряды D0, D1 определяют три разновидности асинхронного режима по частоте сигналов синхронизации, то есть TXC и RXC делятся на 1, на 16 или на 64 и с такой частотой передается информация.

D2, D3 определяют число битов данных передаваемого слова.

Режим контроля задается D4:

D4=0 - контроля нет,

D4=1 - контроль есть.

D5 задает тип контроля.

D6, D7 определяют число передаваемых стоп-битов.

При синхронном обмене данные передаются в виде массива слов, а для синхронизации запуска при приеме данных используется 1 или 2 символа синхронизации.

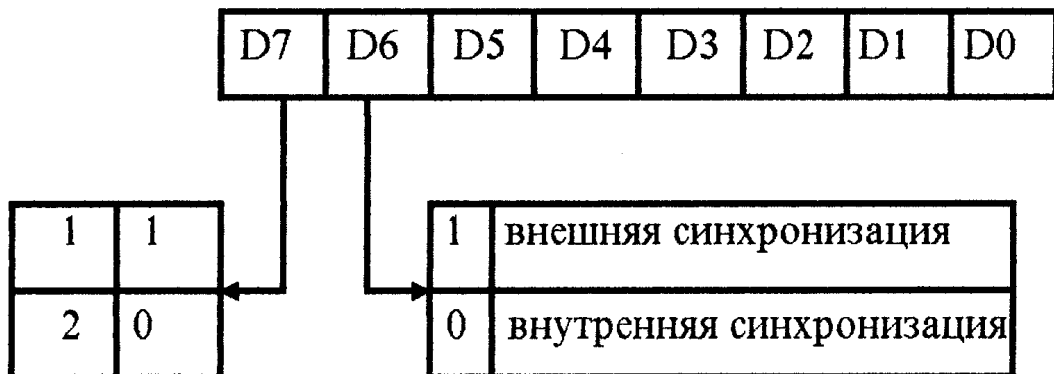


Рис. 2

Формат инструкции синхронного режима представлен на рис. 2.

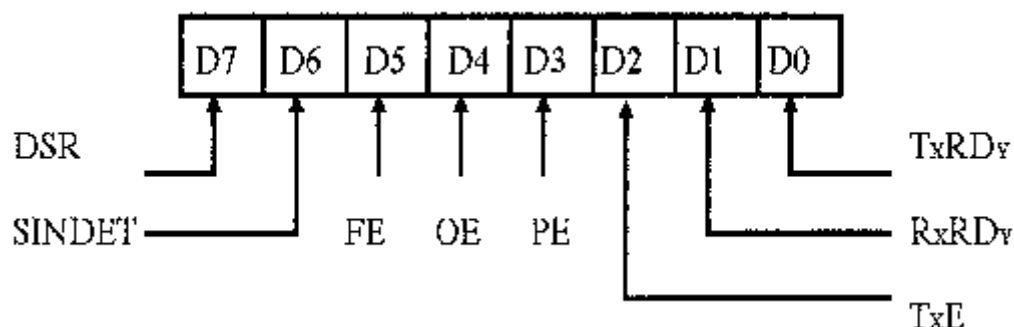
- D0=D1=0 - признак синхронного режима;
D6 - устанавливает вид синхронизации:
 «1» - внешняя (от внешнего синхроимпульса);
 «0» - внутренняя (от внутреннего синхроимпульса);
D7 - определяет число синхросимволов;
D2..D5 - как и у асинхронного обмена.

3.8.3. Команды.

Команды подаются после инструкции режима и управляют выполнением конкретных операций.

<i>Разряды</i>	<i>Назначение</i>	<i>Пояснения</i>
D0	разрешение передачи (TXEN)	передача невозможна при D0=0 и возможна при D0=1
D1	запрос о готовности передатчика терминала к передаче терминала (DTR)	при D1=0 DRT=0 (активный уровень)
D2	разрешение приема (RXE)	при D2=0 прием невозможен при D2=1 прием возможен
D3	конец передачи (SBRK)	D3=0 нормальная работа D3=1 устанавливает сигнал TXD в высокий уровень (TXD=1)
D4	установка ошибок (ER)	D4=1 установка разрядов ошибок в исходное состояние
D5	запрос о готовности приемника терминала к приему (RTS)	D5=1 RTS в активном состоянии равен "0"
D6	программный сброс УСАПП в исходное состояние (IR)	D6=1 устанавливает в исходное состояние
D7	режим поиска синхросимволов (EH)	D7=1 устанавливает режим поиска синхросимволов

3.8.4. Формат слова-состояния.



Кроме уже рассмотренных при описании структурной схемы сигналов, в слове состояния существует три флаговых разряда ошибок:

- PE - устанавливается при ошибке по четности;
- OE - ошибка переполнения, если микропроцессор не успел прочитать слово, а в регистр уже пришло другое;
- FE - ошибка стоп-бита, если не обнаружен стоп-бит.

3.8.5. Режимы работы.

УСАПП может работать в 5 режимах обмена информацией.

1. синхронная передача.
2. синхронный прием с внутренней синхронизацией.
3. синхронный прием с внешней синхронизацией.
4. асинхронная передача.
5. асинхронный прием.

1. Синхронная передача.

При синхронной передаче на выходе TXD с частотой сигнала синхронизации формируется последовательность, начинающаяся с синхросимволов, запрограммированных инструкцией режима. Затем передаются поступающие из МП коды символов, каждый из которых может заканчиваться битом контроля. Если МП не загрузил очередной символ к моменту передачи, то УСАПП вставляет в передаваемую последовательность синхросимволы, а на выходе TXE, вырабатывается сигнал высокого уровня, идентифицирующий пустую передачу.

2. Синхронный прием.

При синхронном приеме с внутренней синхронизацией УСАПП начинает работу с поиска во входной последовательности синхросимволов, сравнивая принимаемые символы с записанными в него синхросимволами.

После обнаружения синхросимволов SINDET переходит в высокое состояние, разрешая прием входных данных.

SINDET сбрасывается автоматически при чтении слова-состояния.

3. Синхронный прием данных с внешней синхронизацией.

При внешней синхронизации на вход SINDET подается сигнал положительного уровня, который разрешает прием данных на вход RXD, со скоростью сигнала синхронизации, поступающем на вход RXC.

Возможна при этом режиме организация приема данных по прерываниям, если

сигнал SINDET используется для запроса прерывания.

4. Асинхронная передача.

Последовательные данные формируются на выходе TXD, по спаду сигнала синхронизации TXC с периодом, задаваемым инструкцией режима и равным 1,16 или 64 периодам сигналов синхронизации (TXC умножаются на 1,16,64 соответственно). Если после передачи символа, следующий символ отсутствует, на выходе TXD устанавливается высокий уровень, до тех пор пока новые данные не поступят от МП. Запись очередного байта УСАПП производится, если в слове-состоянии разряд D0=1, что соответствует высокому уровню на входе TXRDY. Сигнал TXRDY может использоваться для запроса прерывания.

5. Асинхронный прием:

Асинхронный прием начинается с поиска стартового бита, который устанавливает на входы RXD сигнал низкого уровня. Наличие старт-бита вторично проверяется стробированием его середины внутренним строб-импульсом. Если старт-бит найден, то запускается внутренний счетчик битов, который определяет начало и конец битов данных, бит контроля, стоп-биты. Прием стоп-бита идентифицирует окончание приема байта информации. При этом на выходе RXRDY устанавливается «1».

Микропроцессор может считывать информацию по соответствующему флагу в слове-состоянии или по прерыванию, если RXRDY- запрос на прерывание.

3.8.6. Пример программы.

```

0100 MOV     AL,BF
0102 NOP
0103 MOV     DX,027B
0106 OUT     DX,AL
0107 MOV     AL,28
0109 MOV     DX,027A
010C OUT     DX,AL
010D MOV     AL,01
010F OUT     DX,AL
0110 MOV     CX,0002
0113 MOV     DX,02BB
0116 MOV     AL,40
0118 OUT     DX,AL
0119 MOV     DX,02BB
011C MOV     AL,4D
011E OUT     DX,AL
011F NOP
0120 NOP
0121 NOP
0122 MOV     DX,02BB
0125 MOV     AL,33
0127 OUT     DX,AL

```

```

0128 NOP
0129 NOP
012A NOP
012B LOOP 0113
012D MOV DX,02BA
0130 MOV CX,03FF
0133 MOV AL,55
0135 OUT DX,AL
0136 LOOP 0136
0137 JMP 133

```

3.8.7. Осциллограммы на выходе УСАПП:

1) Режим 71h , const AA :

0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---



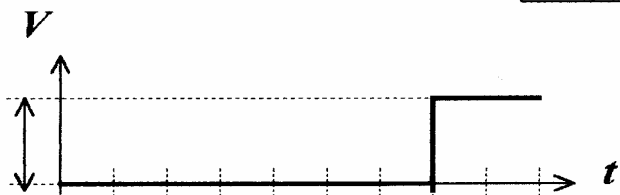
2) Режим 71h , const 55 :

0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---



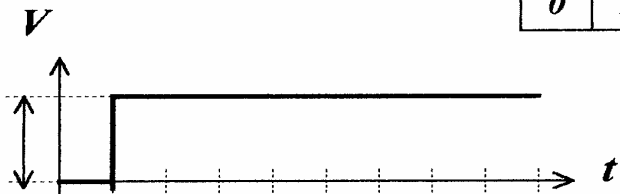
3) Режим 71h , const 00 :

0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---



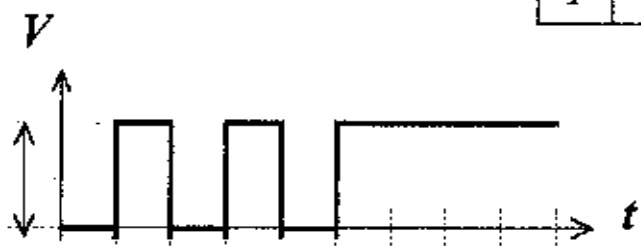
4) Режим 71h , const FF :

0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---



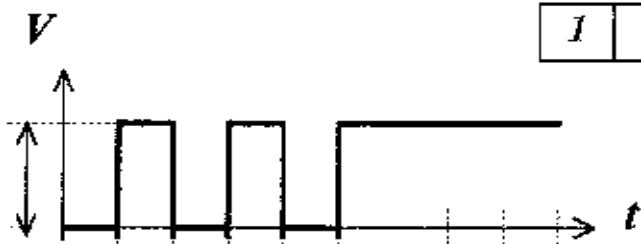
5) Режим B1h, const 55 :

1	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---



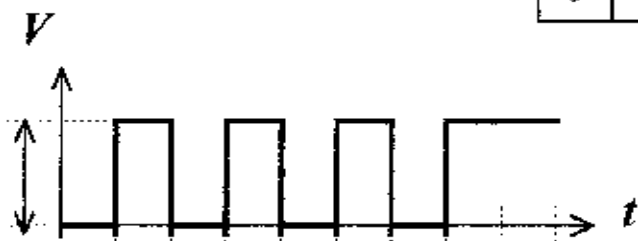
6) Режим F1h, const 55 :

1	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---



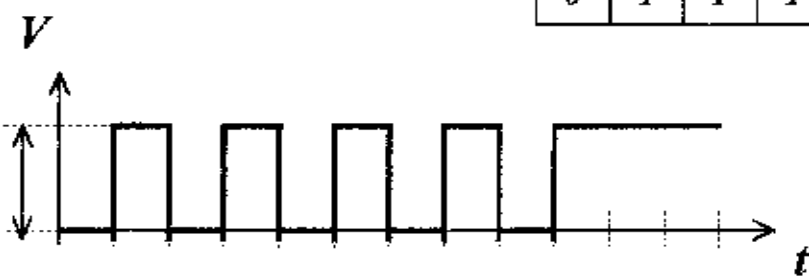
7) Режим 75h, const 55 :

0	1	1	1	0	1	0	1
---	---	---	---	---	---	---	---



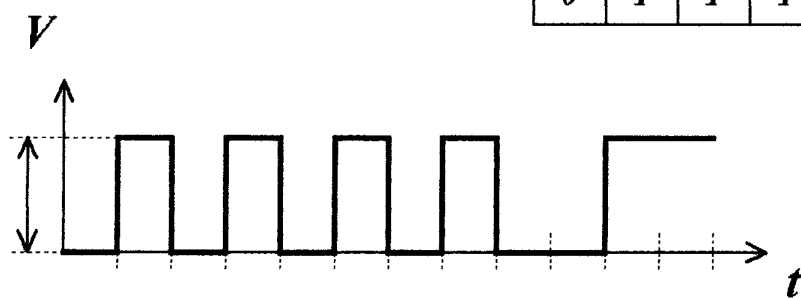
8) Режим 79h, const 55 :

0	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---



9) Режим 7Dh , const 55 :

0	1	1	1	1	1	0	1
---	---	---	---	---	---	---	---



10) Режим 4Dh , const 55 :

0	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---



Лабораторная работа 3 (описание)

4.«БИС параллельного программируемого интерфейса КР580ВВ55 в составе ААПД»

Содержит описание лабораторной работы «БИС программируемого параллельного интерфейса КР580ВВ55 в составе ААПД», проводимой в рамках курса «Адаптеры и контроллеры ЭВМ» с целью изучения принципов работы и программирования программируемого параллельного интерфейса КР580ВВ55, входящего в состав адаптера аппаратуры передачи данных. Предназначено для студентов дневных и вечерних групп.

4.1 . ЦЕЛЬ РАБОТЫ.

Целью лабораторной работы «БИС программируемого параллельного интерфейса КР580ВВ55 в составе ААПД» является ознакомление с принципами работы программного параллельного интерфейса (ППИ) КР580ВВ55, входящего в состав адаптера аппаратуры передачи данных (ААПД) ПЭВМ «Искра 1031», а также с приемами его программной настройки на различные режимы работы.

4.2 . ТЕХНИЧЕСКИЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ В РАБОТЕ.

В лабораторной работе используется ПЭВМ «Искра 1031», ААПД 3.093.142, предназначенный для использования в составе ПЭВМ «Искра 1031», осциллограф С1-81, устройство коммутирующее 5.284.011.

4.3 . ОСНОВНЫЕ ЭЛЕКТРИЧЕСКИЕ ПАРАМЕТРЫ БИС КР580ВВ55.

- | | |
|---|--------------|
| 1) Выходное напряжение логического нуля $U_{OL}, В$ | $< 0,4$, |
| 2) Выходное напряжение логической единицы $U_{OH}, В$ | $> 2,4$, |
| 3) Ток потребления от источника питания $I_{CC}, мА$ | < 60 , |
| 4) Ток утечки каналов А, В, С, D при невыбранном режиме $I_{OZ}, мкА$ | $-100...100$ |
| 5) Ток утечки на управляющих входах $I_L, мкА$ | $-10...10$ |

4.4 . НАЗНАЧЕНИЕ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРУЕМОГО ИНТЕРФЕЙСА КР580ВВ55.

БИС ППИ КР580ВВ55 предназначен для организации ввода-вывода параллельной информации различного формата и позволяет реализовать большинство известных протоколов обмена по параллельным каналам. В ААПД 3.093.138 ППИ используется для ввода в процессор и вывода на стык С2 ряда сигналов серий 100 и 200 и управления режимами работы и таймером. Структурная схема ААПД с входящим в него ППИ приведена на рис.1 ТО ААПД.

4.5 . ФУНКЦИОНИРОВАНИЕ ППИ К580ВВ55.

ППИ К580ВВ55 содержит три независимых порта А, В и С. Режим работы каждого из них программируется с помощью управляющего слова (УС).

УС может задать один из трех основных режимов: основной режим ввода-вывода (режим 0), стробируемый ввод-вывод (режим 1), режим двунаправленной передачи информации (режим 2). Одним УС можно установить различные режимы работы для каждого из каналов. Формат УС

и подробные характеристики режимов см. Приложение 4.9.1.

Канал А может работать в режимах 0,1,2, при этом он установлен аппаратно только на ввод в данном ААПД; канал В - в режимах 0,1; 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.

Программирование ППИ: Чтобы запрограммировать ППИ на определенный режим работы, необходимо занести УС по адресу управляющего регистра Rg упр. В Таблице 1 приведены адреса Rg упр. и регистров данных каналов А,В,С.

Таблица 1. Адреса каналов

	<i>Адрес</i>	<i>Регистр</i>
1	2F8	Порт А
2	2F9	Порт В
3	2FA	Порт С
4	2FB	Rg упр

На электрической принципиальной схеме адаптера АПД 3.093.142.ЭЗ и на структурной схеме ААПД ППИ обозначен как IOP (input output parallel)- параллельный ввод-вывод, построенный на микросхеме D19 (KP580BB55A). На вход поступают сигналы: A0,A1,WR,RD,IOP. На выходе : PA7-PA0,PB7-PB0,PC2,PC3,PC6,PC7.

Выходные сигналы буферизированы - они поступают на входы микросхем D32-D36 (K170АП2). Функции этих микросхем - буферизация сигналов и согласование их по уровням (они выполняют преобразование уровней «1»и «0» ТТЛ -логики в уровни сигналов стандарта С2).

Сигналы PA7-PA2, PB7-PB1 выходят на внешний разъем Х3. На разъем Х3 выходят 27 цепей ,(см. - Схема электрическая принципиальная адаптера АПД 3.093.142 э3). Цепи с номером 1xx - выводы последовательного порта (IOS), цепи с номером -2xx - выводы параллельного порта (IOP). Распределение сигналов по контактам Х3 см. в таблицах 1и 2 технического описания ААПД.

Для практической работы к разъему Х3 подключено коммутирующее устройство 5.284.011 (коннектор).Схема подключения и распределение сигналов по контактам разъема РШ2Н-1-29(С2-100) коннектора приведены в описании лабораторной работы «Универсальный синхронно-асинхронный приемо-передатчик (УСАПП) в составе ААПД...» (рис. 1 и таблица 1).

4.6. УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ

Выполнение лабораторной работы состоит из трех этапов:

- подготовка к лабораторной работе;
- проведение работ по непосредственному изучению параллельного программируемого интерфейса, принципов его работы, программной настройки, написание и отладка программ, которые позволяют проиллюстрировать работу ППИ;
- оформление отчета и сдача (защита) лабораторной работы преподавателю.

4.6.1 Подготовка к лабораторной работе заключается в изучении описания лабораторной работы «БИС программируемого параллельного интерфейса KP580BB55 в составе ААПД», описания работы микросхемы KP580BB55, отладчика AFD, ознакомлении с примерами программ (см. Приложения к лабораторной работе). Подготовка к работе проводится студентом самостоятельно во внеурочное время.

4.6.2 Проведение работ по непосредственному изучению принципов работы ППИ разбивается на два этапа.

На первом этапе определяются управляющие слова для

различных режимов работы ППИ, адреса регистров ППИ. На этом этапе необходимо выяснить, какие выводы ППИ выходят на разъемы коннектора, какие выводы ППИ аппаратно соединяет заглушка.

На втором этапе проводится создание и отладка программ, иллюстрирующих возможности программирования ППИ.

4.6.3 Оформление отчета по лабораторной работе осуществляется во внеаудиторное время. Сдача (защита) лабораторной работы производится во время выполнения следующей работы или во внеурочное время по согласованию с преподавателем.

4.6.4 Отчет по лабораторной работе должен содержать следующий материал:

- постановку задачи выполняемой лабораторной работы;
- краткое описание режимов работы ППИ и управляющие слова для каждого режима;
- результаты изучения распределения выводов ППИ по контактам разъема коннектора;
- тексты программ на ассемблере;
- осциллограммы электрических сигналов;
- выводы о проделанной работе.

Отчет должен быть оформлен грамотно и аккуратно.

4.7. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ.

4.7.1 По описанию лабораторной работы определить адреса портов А, В, С и регистра управления.

4.7.2 Определить управляющие слова для различных режимов работы ППИ:

- 1) режим 0 : канал А- ввод, канал В- вывод, канал С: PC7-PC4-вывод, PC3-PC0-ввод
- 2) режим 0 : канал А- ввод, канал В- ввод, канал С: PC7-PC4-ввод, PC3-PC0-ввод
- 3) режим 0 : канал А- ввод, канал В- вывод, канал С: PC7-PC4-вывод, PC3-PC0-вывод.

4.7.3 Пользуясь командами отладчика AFD I и O, определить, какие выводы канала В ППИ выходят на разъем коммутирующего устройства. Для этого записывать поочередно константы 01,02,04 ...80 (бегущую единицу) по адресу порта В, при этом проверяя состояния выводов по осциллографу. Распределение сигналов приведено в таблице 3 приложения.

4.7.4 Установить в разъем коннектора C200 заглушку, аппаратно коммутирующую входы РВ на входы РА. Выяснить, какие именно выводы соединены друг с другом. Для этого запрограммировать порт В на вывод, режим 0; порт А -на ввод, режим 0, а затем записать поочередно в порт В константы 01,02,04 ...80 (бегущую единицу), при этом сравнивая содержимое портов А и В (с помощью команды IN). Если выходы соединены, то в разряде, соответствующем этим выходам, биты будут совпадать.

4.7.5 Сделать генератор импульсов на выводе РВ3 с положительным полупериодом, равным 40 мкс и отрицательным полупериодом, равным 10 мкс, используя в качестве примера программу № 2 Приложения 4.9.2. Осциллограмму зарисовать.

4.7.6 Написать программу, реализующую генератор импульсов на выводах РВ3, РВ4, РВ5, РВ6 таким образом, чтобы частота на каждом выходе этих разрядов отличалась в два раза. Осциллограммы зарисовать.

4.7.7 Определить длительность цикла LOOP (с помощью осциллографа).

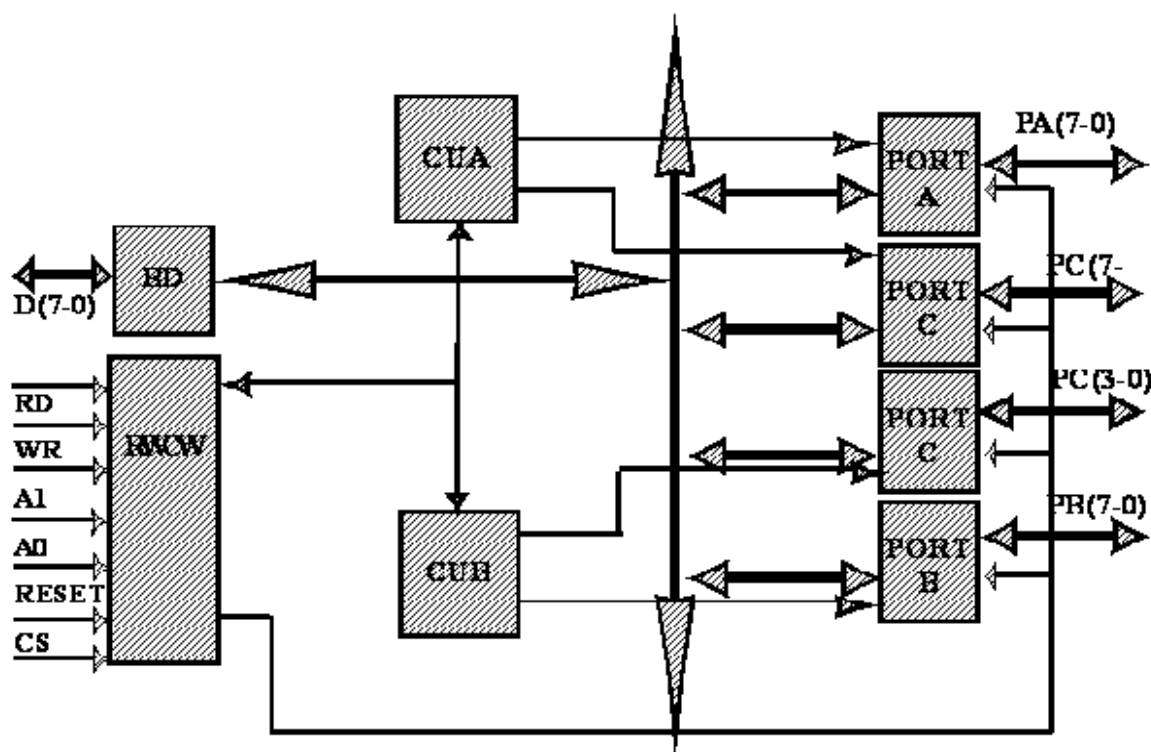
4.7.8 Определить максимальную частоту программного генерирования сигнала на выходах порта В. Для этого в порт В (например, в разряд PB4) заносить программно сначала «1», затем «0» в цикле. Осциллограмму зарисовать. Объяснить, почему положительный и отрицательный полупериоды не одинаковы.

4.8. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

- 1) Справочник «Микропроцессоры и микроЭВМ в системах автоматического управления» под редакцией С.Т. Хвоща. Л., «Машиностроение», 1987 г.
- 2) Лекции по курсу «Адаптеры и контроллеры ЭВМ»
- 3) Адаптер АПД. Техническое описание 3.093.138 Т0

4.9. ПРИЛОЖЕНИЯ

4.9.1. Описание ППИ КР580ВВ55:



Структурная схема ППИ

В состав ППИ входят :

- двунаправленный восьмиразрядный буферный регистр данных (BD), связывающий ППИ с системной шиной данных ;
- блок управления чтением/записью (RWCW), обеспечивающий управление внешней и внутренней передачей данных , управляющих слов и информации о состоянии ППИ ;
- три восьми разрядных канала (PORT A , PORT B , PORT C), предназначенных для обмена информацией с внешними устройствами ;
- схема управления группой A (CUA), вырабатывающая сигналы управления группой A и старшими разрядами канала C ;
- схема управления группой B (CUB), вырабатывающая

сигналы управления группой В и младшими разрядами канала С .

Сигналы ППИ :

- D (7-0) - вход/выход данных со стороны системной шины ;
- RD - чтение ; низкий уровень сигнала разрешает считывание информации из регистра , адресуемого по входам A0 , A1 на шину D (7-0) ;
- WR - запись ; низкий уровень сигнала разрешает запись информации с шины D (7-0) в регистр , адресуемый по входам A0 , A1 ;
- A0 , A1 - входы адресации внутренних регистров ППИ ;
- RESET - сброс ; высокий уровень сигнала обнуляет регистр управляющего слова и устанавливает все порты в режим ввода ;
- CS - выбор микросхемы ; низкий уровень сигнала подключает ППИ к системной шине ;
- PA (7-0) - входы/выходы канала А ;
- PB (7-0) - входы/выходы канала В ;
- PC (7-4) и PC (3-0) входы/выходы канала С .

Сигналы управления работой ППИ подаются на бок RWCU задавая вид операции, выполняемой ППИ (см. Табл. 2).

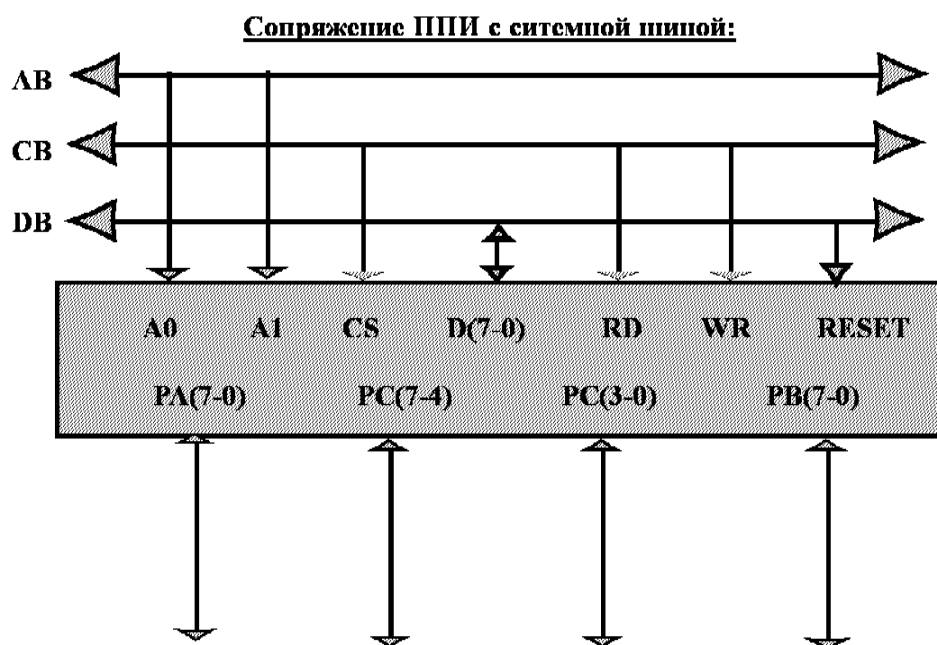


Таблица 2.

Операция	CS	RD	WR	A0	A1
Запись управляющего слова из МП	0	1	0	1	1
Запись в регистр канала А	0	1	0	0	0
Запись в регистр канала В	0	1	0	0	1
Запись в регистр канала С	0	1	0	1	0
Чтение из канала А	0	0	1	0	0
Чтение из канала В	0	0	1	1	1
Чтение из канала С	0	0	1	0	0
Отключение ППИ от шины	1	x	x	x	x

Функционирование :

Режим работы каждого из каналов ППИ программируется с помощью управляющего слова . Оно может задавать один из трех режимов :

- основной режим ввода/вывода (режим 0) .
- стробируемый ввод/вывод (режим 1) .
- режим двунаправленной передачи данных (режим 2) .

Одним управляющим словом можно установить различные режимы работы для каждого из каналов .

Формат управляющего слова.

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
-----	-----	-----	-----	-----	-----	-----	-----

Значения разрядов :

- D7 : 1 - установка режима ;
0 - сброс/установка отдельных разрядов канала С .
- D6 , D5 : 0 , 0 - канал А работает в режиме 0 ;
0 , 1 - канал А работает в режиме 1 ;
1 , 0 - канал А работает в режиме 2 .
- D4 : 1 - канал А работает в режиме ввода ;
0 - канал А работает в режиме вывода .
- D3 : 1 - (7-4) линии канала С работают в режиме ввода ;
0 - (7-4) линии канала С работают в режиме вывода .
- D2 : 1 - канал В работает в режиме 0 ;
0 - канал В работает в режиме 1 .
- D1 : 1 - канал В работает в режиме ввода ;
0 - канал В работает в режиме вывода .
- D0 : 1 - (3-0) линии канала С работают в режиме ввода ;
0 - (3-0) линии канала С работают в режиме вывода .

Канал А может работать в трех режимах , канал В - в режимах 0 , 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 .

Разряд 7 управляющего слова определяет либо установку режимов каналов (D7=1), либо работу ППИ в режиме сброса/установки отдельных разрядов канала С (D7=0) .

При поразрядном управлении каналом С разряды D(3-1) определяют номер модифицируемого разряда , а нулевой разряд задает сброс (D0=0) или установку (D0=1) этого разряда . D (6-4) в этом случае не используются . Сброс/установку разрядов канала С можно использовать для разрешения выработки сигналов запроса прерываний от ППИ .

Для каждого из каналов А и В в ППИ имеется триггер разрешения прерывания , установка/сброс которого осуществляется управляющим словом установки/сброса определенного разряда канала С . Если триггер разрешения прерывания соответствующего канала установлен (INTE=1) , то ППИ может сформировать сигнал запроса прерывания при готовности внешнего устройства к вводу или выводу .

Режим 0 - применяется при синхронном обмене или при программной организации асинхронного обмена . В этом режиме микросхема может рассматриваться как устройств, состоящее из четырех портов (два восьмиразрядных и два четырехразрядных) . Независимо друг от друга они настраиваются на ввод или вывод . Вывод осуществляется по команде OUT МП с фиксацией выводимой информации в регистрах каналов . Ввод по команде IN без запоминания

информации .

Режим 1 - обеспечивает стробируемый однонаправленный обмен информацией с внешним устройством . Передача данных производится по каналам А и В , а линии канала С управляют передачей . Работу канала в режиме 1 сопровождают три управляющих сигнала . Если один из каналов запрограммировать на режим 1 , то остальные тринадцать интерфейсных линий могут быть использованы в режиме 0 . Если оба канала запрограммированы на режим 1 , то оставшиеся две интерфейсные линии канала С могут быть настроены на ввод или вывод .

В режиме 1 для ввода информации используются следующие управляющие сигналы :

- STB - строб приема; входной сигнал, формируемый внешним устройством (ВУ) и указывающий на готовность ВУ к вводу информации
- IBF - подтверждение приема ; выходной сигнал ППИ , сообщающий ВУ об окончании приема данных в канал . Он формируется по срезу STB .
- INTR - запрос прерывания ; выходной сигнал ППИ , информирующий МП о завершении приема информации в канал . Высокий уровень сигнала устанавливается при STB=1 , IBF=1 , INTE=1 . Сбрасывается спадом сигнала RD . Для операции ввода управление сигналом INTE канала А осуществляется по линии PC4 , а канала В - по линии PC2 .

Для вывода информации используются управляющие сигналы :

- OBF - строб записи ; выходной сигнал , указывающий ВУ о готовности к выводу . Формируется по фронту WR .
- ACK - подтверждение записи ; входной сигнал от ВУ , подтверждающий прием информации из ППИ .
- INTR - запрос прерывания ; выходной сигнал ППИ , информирующий МП о завершении операции вывода информации . Высокий уровень устанавливается по фронту сигнала ACK =1 при INTE=1 . Сбрасывается спадом сигнала WR. При операции вывода управление сигналом INTE каналом А осуществляется по линии PC6 , а канала В - по линии PC2 .

На рисунках показан ППИ , работающий в режиме 1 , и соответствующее ему управляющее слово для ввода информации по каналам А и В (рис. а) и для вывода (рис.б) .

Неиспользуемые для передачи управляющих сигналов линии PC6,7 (рис.1 а) и PC5,4 (рис.1 б) могут быть запрограммированы на ввод (D3=1) или вывод (D3=0) .

На рис.1 в приведен вариант конфигурации ППИ в режиме 1 для вывода по каналу А и ввода по каналу В .

Режим 2 - обеспечивает двунаправленную передачу информации по каналу А к внешнему устройству и обратно . Процесс обмена сопровождают пять управляющих сигналов , подаваемых по линиям PC(7-3) . Оставшиеся одиннадцать интерфейсных линий могут настраиваться на режим 0 или режим 1 . Распределение сигналов по интерфейсным линиям и управляющее слово показаны на рис.1г:

Разряд D0 определяет настройку на ввод или вывод интерфейсных линий PC(2-0) . Функции управляющих сигналов аналогичны режиму 1 . Управление установкой внутреннего сигнала INTE для операции ввода осуществляется по линии PC4 , а для вывода - по линии PC6 .

В режимах 1 и 2 возможно проведение контроля за состоянием работы ВУ и ППИ . Контроль осуществляется чтением слова состояния канала С по команде IN . Форматы слова состояния приведены на рис.2 :

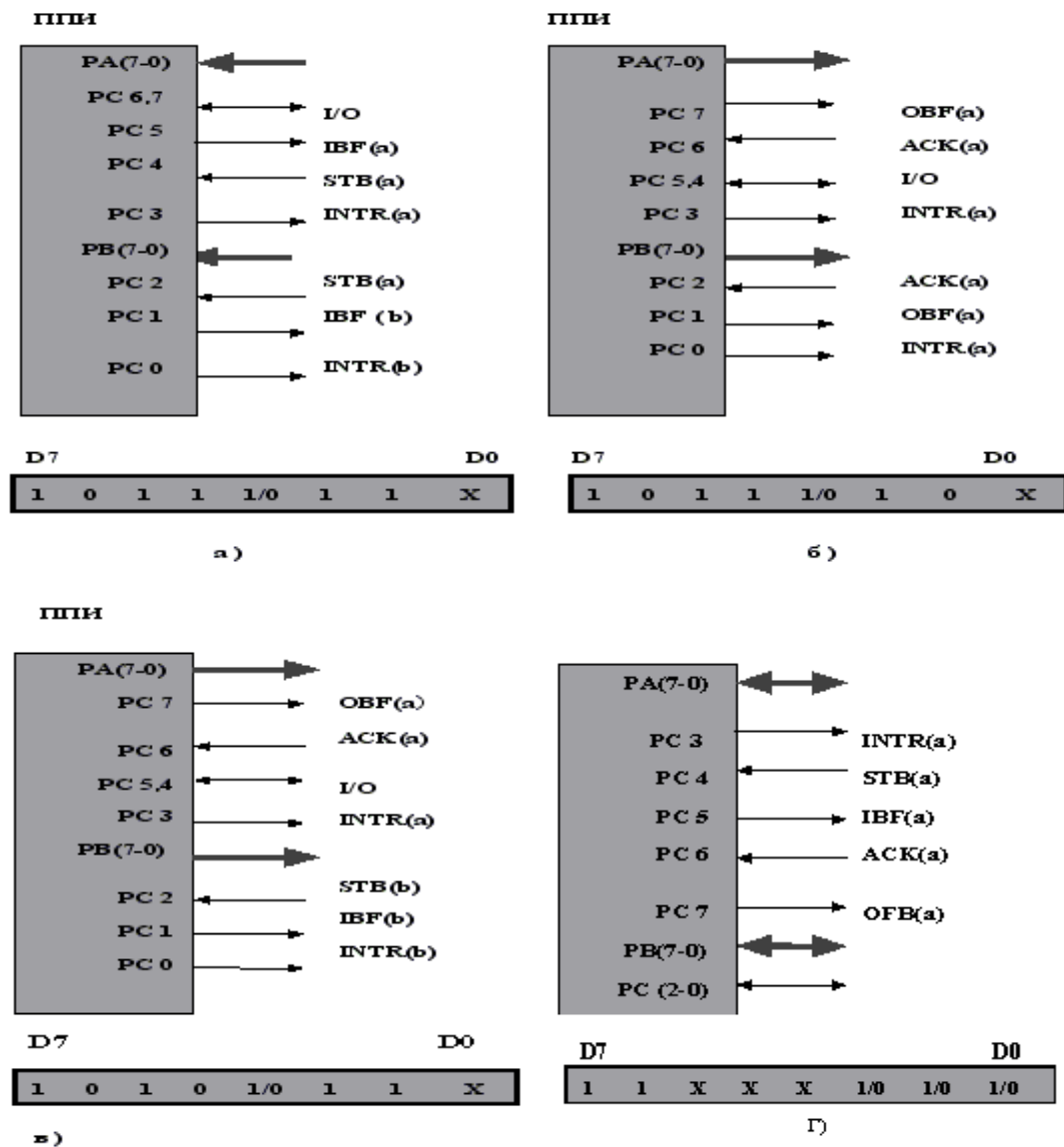


Рис. 1

Режим 1 (ВВОД):

I/O I/O IBF(a) INTE(a) INTR(a) INTE(b) IBF(b) INTR(b)

Режим 1 (ВЫВОД):

OBF(a) INTE(a) I/O I/O INTR(a) INTE(b) OBF(b) INTR(b)

Режим 2:

OBF(a) INTE(a) IBF(a) INTE(a) INTR(a) X X X

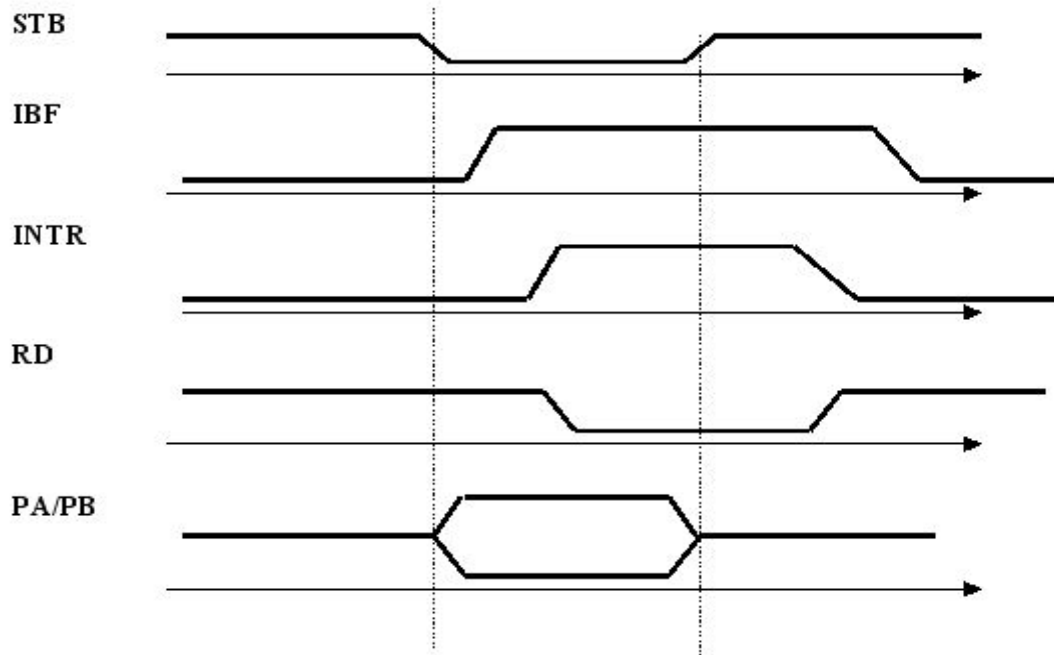
Рис. 2

Для режима 1 сигналы ввода/вывода в разрядах с определенными номерами указывают на операцию ввода или вывода по интерфейсным линиям канала С с теми же номерами .

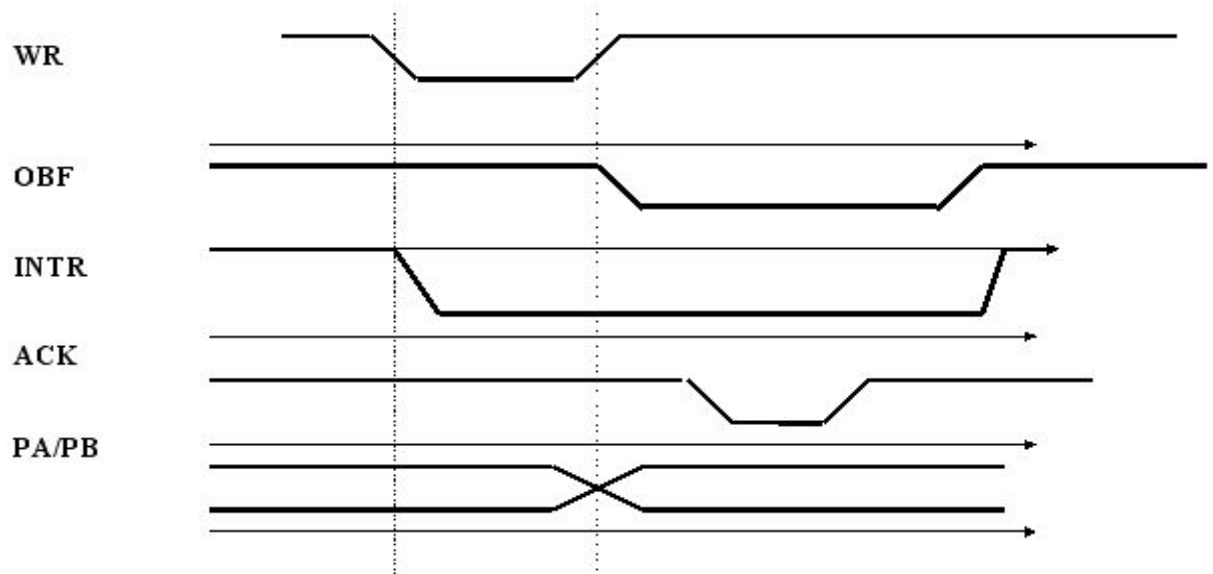
Для режима 2 значения разрядов D(2-0) определяются только режимом работы группы В .

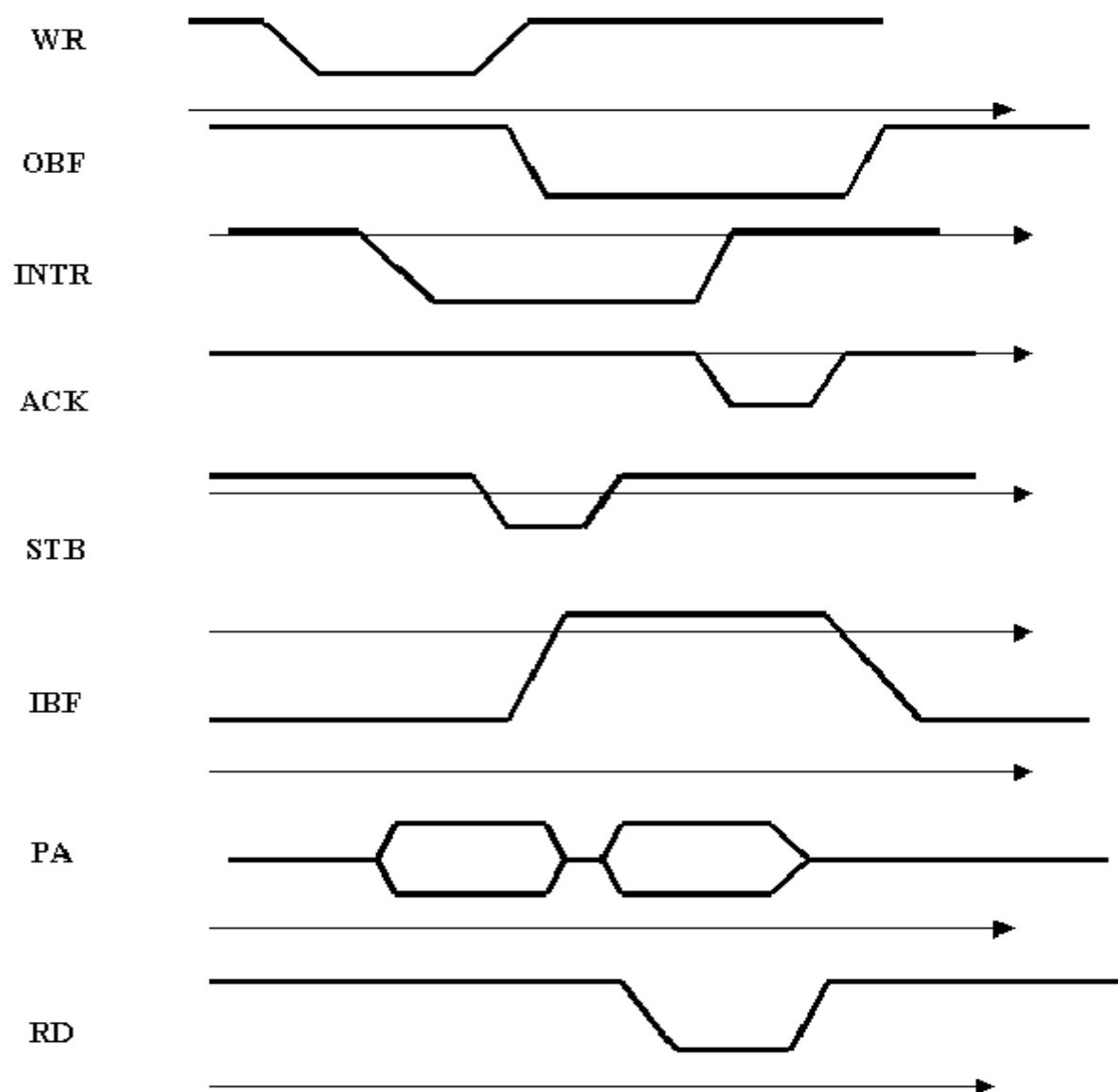
Временные диаграммы работы ППИ в режиме 1.

Ввод:



Вывод:



Временная диаграмма работы ППИ в режиме 2:

4.9.2 Примеры программ :

Программа 1.

```

mov DX,2FB          - программирование ППИ
mov AL,99
out DX,AL
mov DX,2F9          - занесение в порт константы AA
mov AL,AA
out DX,AL
in  AX,2F9          -занесение содержимого порта В в регистр AX
int 3

```

Программа 2.

Генератор меандра на выходе выхода PB3 (контакт 9):

```

      mov      DX,2FB      - программирование ППИ
      mov      AL,99
      out      DX,AL
      mov      DX,2F9      - занесение в порт В константы 00
A1:   mov      AL,00
      out      DX,AL
      mov      CX,FFF      -задержка . При CX= FFFh период  равен 25мс.
A2:   loop     A2
      mov      AL,04
      out      DX,AL
      mov      CX,FFF
A3:   loop     A3
      jmp      A1
      int 3

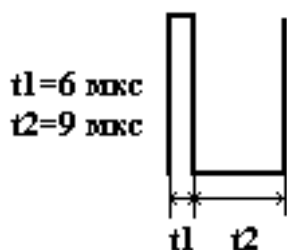
```

Программа 3. Определение максимальной частоты.

```

      MOV  DX,02FB
      MOV  AL,9D
      OUT  DX,AL
      MOV  DX,02F9
A1:    MOV  AL,00
      OUT  DX,AL
      MOV  AL,10
      OUT  DX,AL
      JMP  A1:

```



Максимальная частота = $1/(6+9) = 67 \text{ КГц}$

4.9.3. Распределение сигналов по контактам разъема коннектора.

Таблица 3.

Контакт	Цепь	Выводы ППИ
1	Цепь 101(201)	
2	Цепь 103(203)	РА6, ВХ
3	Цепь 104(204)	РА5, ВХ
4	Цепь 105(205)	РА4, ВХ
5	Цепь 106(206)	РВ5, ВЫХ
6	Цепь 107(207)	РВ4, ВЫХ
7	Цепь 102(202)	РВ7, ВЫХ
8	Цепь 109(209)	РВ2, ВЫХ
9	Цепь 108(208)	РВ3, ВЫХ
10	Цепь 114(210)	РА3, ВХ
11	Цепь 126	РВ1, ВЫХ
12	Цепь 115(211)	РВ6, ВЫХ
13	Цепь 120(213)	РА7, ВХ
14	Цепь 125	РА2, ВХ

Цепи. с номерами, начинающимися с 1, соответствуют разъему «С2(100)», а начинающиеся с 2 - разъему «С2(200)».

ЛАБОРАТОРНАЯ РАБОТА 4 **(описание)**

5. « ПАРАЛЛЕЛЬНЫЙ ПОРТ IBM/PC AT »

5.1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы « Параллельный порт IBM/PC AT через BIOS » является ознакомление с принципами работы двунаправленного параллельного порта и принципами его программирования через функции BIOS , а также с приемами непосредственной его программной настройки на выдачу данных.

5.2. ТЕХНИЧЕСКИЕ СРЕДСТВА , ИСПОЛЬЗУЕМЫЕ В РАБОТЕ

В лабораторной работе используется IBM PC/AT , осциллограф C1-81.

5.3. ПРОГРАММИРОВАНИЕ ПАРАЛЛЕЛЬНОГО ПОРТА

Временная диаграмма параллельного порта определяется временной диаграммой подключенного устройства . Если подключенное устройство стандартное , то достаточно воспользоваться прерыванием INT 17H (ее функциями).

Примечание : считывание значения регистра состояния порта нужно проводить дважды из-за затянутости фронта сигнала BUSY.

Двунаправленный порт , как правило по умолчанию , устанавливается в режим совместимости. В ЭВМ PS/2 имеется программная возможность управлять режимом работы двунаправленного порта , путем записи по адресу 0102h значения в седьмой разряд : 0 - расширенный режим , 1 - совместимый .

В некоторых современных ПЭВМ тип параллельного порта выбирается в SETUP-е (стандартный , ECP, EPP, EPP+ECP).

5.4. ПРЕРЫВАНИЯ ПРИНТЕРА

В BIOS предусмотрены функции для инициализации принтера , печати символа , и чтения состояния принтера.

Прерывания принтера IRQ (7) доступно системной программе.

При POST BIOS ищет параллельные интерфейсы по трем стандартным портам : 3BCh, 278h , 378h.

Когда интерфейс найден , BIOS помещает базовый адрес порта принтера в область памяти BIOS , начиная с адреса 0 : 0408h и , следовательно , при программировании параллельного порта следует пользоваться не жестко привязанным адресом , а адресом , извлеченным из памяти BIOS.

BIOS принтера предусматривает так же функцию тайм-аута , длина которого программируется пользователем. Для этого используются адреса 0 : 0478h-0 : 047Bh. Значения тайм-аута устанавливаются POST на 20 , и могут быть установлены в диапазоне от 1 до 255. Каждая единица соответствует примерно одной секунде.

5. 5. ПРЕРЫВАНИЯ ПРИНТЕРА ЧЕРЕЗ ФУНКЦИИ BIOS-а.

INT 05H - печать копии экрана (программное) ;

INT 0FH - IRQ7-прерывание от принтера (аппаратное), по сигналу ACK.

INT 17H - драйвер принтера (программное)

В таблице 1 приведены адреса ячеек памяти , в которые помещаются вектора прерываний и другая информация BIOS для параллельного порта.

Таблица 1.

Адрес	Число Байт	Функция
0 : 0014h	4	Вектор прерывания INT 05h
0 : 0036h	4	Вектор прерывания INT 0Fh
0 : 0056h	4	Вектор прерывания INT 17h
0 : 0408h	2	Адрес базового порта LPT1
0 : 040Ah	2	Адрес базового порта LPT2
0 : 040Ch	2	Адрес базового порта LPT3
0 : 040Eh	2	Адрес базового порта LPT4
0 : 0478h	1	Тайм-аут LPT1
0 : 0479h	1	Тайм-аут LPT2
0 : 047Ah	1	Тайм-аут LPT3
0 : 047Bh	1	Тайм-аут LPT1
0 : 0500h	1	Состояние для функции "печать экрана"

5.6. ДРАЙВЕР ПРИНТЕРА INT 17h.

Вызывается для всех функций , относящихся к драйверу принтера :

00h- печать символа;

01h- инициализация;

02h- определение состояния

Функция 00h - печать символа.

Вход : AH= 00h

AL= символ для печати (любой из набора 00h-FFh)

DX= № принтера (№ порта от 0 до 3).

Выход : AH= состояния : D0 - состояние тайм-аута ;

D1, D2 - не используется ;

D3 - ошибка ввода/вывода ;

D4 - выбран ;

D5 - нет бумаги (Paper End);

D6 - подтверждение (ACK);

D7 - готов.

Если принтер не готов , то при выполнении INT 17h с AH=00h выполняется прерывание INT 15h с AH=90h которое называется «устройство занято».

Функция 01h - инициализация.

Вход : AH= 01h

DX= № принтера (№ порта от 0 до 3).

Выход : AH= состояния принтера (см. функцию 00h).

Функция 02h -определение состояния.

Вход : AH= 02h

DX= № принтера (№ порта от 0 до3).

Выход : AH= состояния принтера (см. функцию 00h).

В таблице 2 приведены характерные состояния возврата.

Таблица 2.

А Н	О п и с а н и е
00h	Режим "off-line"
02h	Не установлен порт принтера
08h	Питание выключено
30h	Отключен кабель
90h	Прочитать состояние принтера

5.7. УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ

Выполнение лабораторной работы состоит из трех этапов :

- ⌚ подготовка к лабораторной работе ;
- ⌚ проведение работ по непосредственному изучению двунаправленного параллельного порта IBM/PC AT , принципов работы и программирования порта через BIOS , которые позволяют проиллюстрировать его работу.
- ⌚ оформление отчета и защита лабораторной работы преподавателю.

5.7.1 Подготовка к лабораторной работе заключается в изучении описания лабораторной работы « Выдача данных через параллельный порт IBM/PC AT через BIOS в среде AFD», описания контроллера параллельного порта , отладчика AFD , ознакомление с примером программы и осциллограммами (см. Приложения). Подготовка к работе проводится студентом самостоятельно во внеурочное время.

5.7.2 Проведение работ по непосредственному изучению принципов работы двунаправленного параллельного порта разбивается на два этапа.

На первом этапе изучаются структурная схема порта , описание его регистров , прерывания принтера через функции BIOS-а.

На втором этапе проводится создание и отладка программ , демонстрирующих возможности программирования порта.

5.7.3 Оформление отчета по лабораторной работе осуществляется во внеаудиторное время. Сдача (защита) лабораторной работы производится во время выполнения следующей работы или во внеурочное время по согласованию с преподавателем.

5.7.4 Отчет по лабораторной работе должен содержать следующий материал :

- постановку задачи выполняемой лабораторной работы;
- краткое описание работы параллельного порта;
- тексты программ на ассемблере;
- осциллограммы;
- выводы о проделанной работе.

Отчет должен быть оформлен грамотно и аккуратно.

5.8. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

5.8.1 Программным путем осуществить инициализацию порта.

5.8.2 Выяснить, исходя из структурной схемы порта (см. Приложения) и протокола интерфейса Centronics, какие из его выводов необходимо удерживать в верхнем, а какие в нижнем состоянии.

5.8.3 Сделать генератор импульсов на выводе D0 с положительным полупериодом, равным 10 мкс и отрицательным полупериодом, равным 2,5 мкс, используя в качестве примера программу 2 Приложения.

Осциллограмму зарисовать. То же сделать с использованием функций прерывания INT 17h.

5.8.4 Написать программу и подобрать константы для получения на D1, D2, D3, D4 с частотами, отличающихся друг от друга в два раза, основываясь на программе, указанной в Приложении. Осциллограмму зарисовать. То же сделать с использованием функций прерывания INT 17h.

5.8.5 Определить длительность цикла LOOP. Осциллограмму зарисовать.

5.8.6 Замерить возможную минимальную длительность генерируемых сигналов (без цикла LOOP). Осциллограмму зарисовать. То же сделать с использованием функций прерывания INT 17h.

5.9. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

- 1) Лекции по курсу «Адаптеры и контроллеры ЭВМ»
- 2) «Аппаратура персональных компьютеров и ее программирование IBM PC/XT/AT и PS/2». М: Радио и связь, 1995 г.
- 3) Гук М. «Аппаратные средства IBM PC», издательство «Питер Ком», 1998 г.

5.10. ПРИЛОЖЕНИЯ

5.10.1 Описание контроллера параллельного порта.

Ранее использовался однонаправленный порт для подключения принтера.

В PC/2 и некоторых AT-совместимых компьютерах параллельный порт, помимо обычного режима (режима совместимости), поддерживает расширенный режим, который может работать в двунаправленном вводе/выводе.

BIOS поддерживает до трех параллельных портов, которые определяются на этапе POST (тесты). В таблице 3 приведены номера портов и соответствующие им 16-ти разрядные адреса базовых регистров.

Таблица 3.

Порт	Регистр данных	Регистр состояния	Регистр управления
LPT1	3BCh	3BDh	3BEh
LPT2	378h	379h	37Ah
LPT3	278h	279h	27Ah

При обнаружении соответствующего порта BIOS пишет адрес его регистра данных, начиная с адреса 0 : 0408h и присваивает ему имя LPT (n=1,2,3).

BIOS понимает так же имя LPT4, но для работы с ним программист должен сам записать соответствующий адрес в ячейку памяти с адресом 0 : 0410h.

5.10.2 Структурная схема двунаправленного параллельного порта (рис. 1).

5.10.3 Описание регистров параллельного порта.

Регистр данных для чтения/записи байта данных. В режиме совместимости запись данных приводит их к немедленной передаче в линию. Передача данных в двунаправленном режиме управляется путем записи бита направления передачи в регистр управления. Только при бите управления, равного 0, данные передаются в линию. Операция чтение регистра данных приводит к чтению последнего записанного в него значения в режиме совместимости или передаче в двунаправленном режиме. При выполнении чтения при приеме в

двунаправленном режиме (бит направления равен 1) из регистра считывается значение линии.

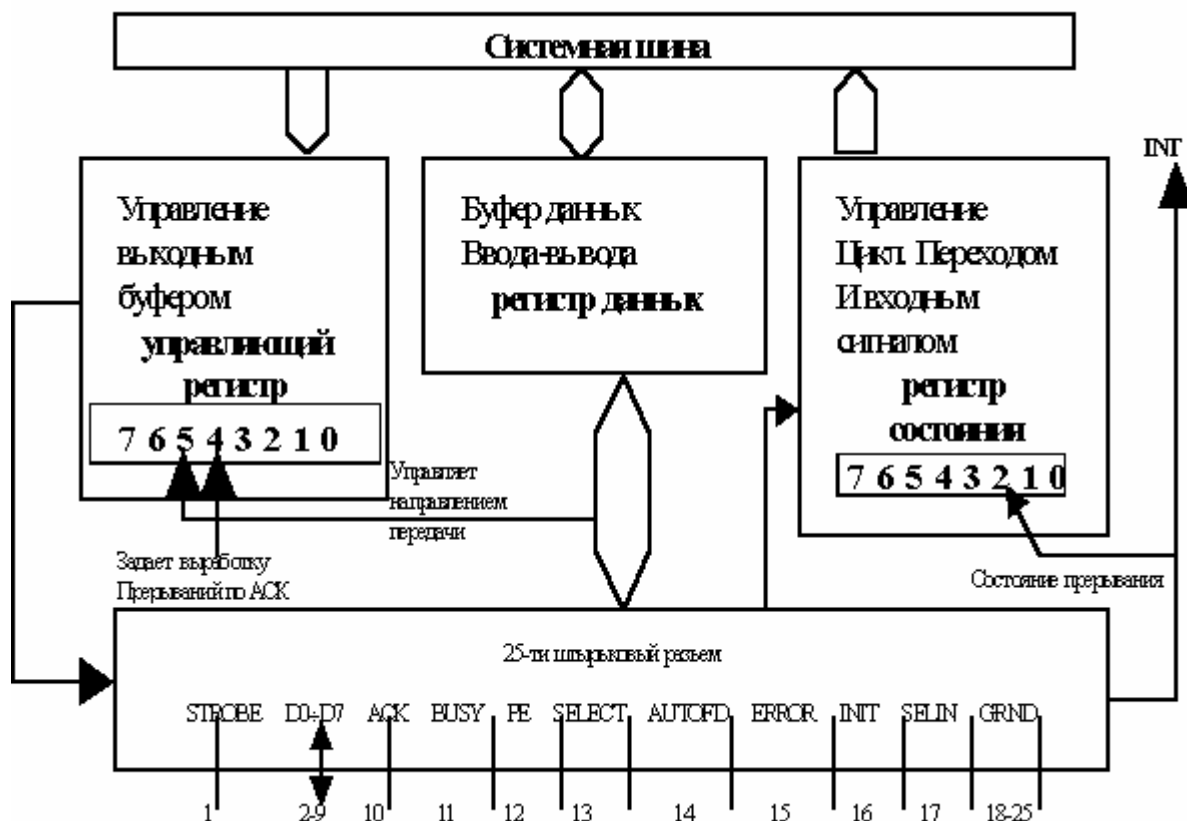


Рис.1.

Структурная схема двунаправленного параллельного порта.

Регистр состояния доступен только для чтения. Служит для получения информации о работе устройства. Считываемое из регистра значение позволяет определить уровень сигнала, на некоторых линиях, управляющих подключенным устройством. Контроллер может вырабатывать прерывание INT для процессора при получении от устройства сигнала подтверждения приема символа + ACK. Такая возможность управляется 4-ым битом регистра управления. Формат регистра состояния приведен в таблице 4.

Таблица 4.

7	6	5	4	3	2	1 - 0
BUSY	ACK	PE	SEL	ERR	IRQS	PE3EPB

BUSY- передает инвертируемое состояние « линия занята»;

ACK- показывает инвертируемое состояние линии готовности к приему очередного байта (подтверждение приема);

PE (Paper end)- показывает текущий сигнал от принтера о состоянии бумаги, 1- нет бумаги;

SEL- показывает текущее состояние сигнала « выборка», 1- устройство выбрано;

ERR- состояние линии ошибки;

IRQS- устанавливается в 0, когда устройство подтвердило

прием предыдущего байта информации (т.е ACK=1). На основании этого сигнала вырабатывается сигнал « запрос на прерывание»-INT.

Регистр управления доступен для записи и чтения в любом из режимов. Используется для задания режима работы контроллера и для передачи в линию ряда управляющих сигналов. Формат регистра управления приведен в таблице 4.

Таблица 4.

7-6	5	4	3	2	1	0
PE3EPB	DIR	IRQE	SELIN	INIT	AFD	STRB

DIR- используется для задания типа операции при работе в расширенном режиме. Направление передачи данных (от машины к устройству или наоборот): 0- операция записи (к ВУ), 1- чтение (от ВУ). В режиме совместимости или для обычного порта, значение этого бита игнорируется.

IRQE- управляет прерыванием; 1- разрешено прерывание по ACK.

SELIN- управляет состоянием сигнала выборки устройства; 1- устройство выбрано.

INIT- управляет инвертируемым сигналом инициализации устройства; 0- принтер инициализировался.

AFD (AutoFD)- управляет состоянием сигнала автоматического прогона строки; 1- принтер после печати каждой строки будет автоматически переходить на новую строку.

STRB- управляет синхронной передачей данных в устройство (т.е сигналом Strobe).

Для подключения внешних устройств используется 25-контактный разъем с разводкой сигналов, согласно интерфейсу Centronics. В этих габаритах могут использоваться двунаправленный интерфейс типа 1, интерфейс с DMA типа 3, а также расширенный интерфейс EPP, ECP.

В Таблице 5 приведено распределение сигналов по контактам разъема в стандартном режиме и при использовании EPP(Enhanced parallel port).

Таблица 5.

Контакт 25 шт.	DIR (направл.)	STANDARD	EPP	Контакт на принтере 36 шт.
1	I/O	-Strobe	-Write	1
2	I/O	Data 0	Data 0	2
3	I/O	Data 1	Data 1	3
4	I/O	Data 2	Data 2	4
5	I/O	Data 3	Data 3	5
6	I/O	Data 4	Data 4	6
7	I/O	Data 5	Data 5	7
8	I/O	Data 6	Data 6	8
9	I/O	Data 7	Data 7	9
10	I/O	-ACK	-IRQ	10
11	I/O	BUSY	WAIT	11
12	I/O	PE	PE	12
13	I/O	SELECT	SELECT	13
14	I/O	-AutoLF	DataSTB	14
15	I/O	-ERROR	-ERROR	32
16	I/O	-INIT	-INIT	31
17	I/O	-SELECT	-AddeSTB	36
18-25	-	GND земля	GND земля	19-30,33

5.10.4 Примеры программ.

Примечание: в программах присутствуют некоторые неточности, которые необходимо устранить в процессе их отладки

Программа 1.

```

MOV AH,01      -инициализация порта
MOV DX, 0      -номер порта
INT 17
A:  MOV AH,00   -печать символа
    MOV AL, 01  -вывод D0
    INT 17
A1: MOV CX,1    -задержка. При CX=0001h длительность
    LOOP A1     импульса равна 0.1 мс.
    MOV AH,00
    MOV AL, 02  -вывод D1
    INT 17
A2: MOV CX,FF   -задержка. При CX=FFh длительность
    LOOP A2     импульса равна 0.2 мс.
    MOV AH,00
    MOV AL, 04  -вывод D2
    INT 17
A3: MOV CX,2F0  -задержка. При CX=2F0h длительность
    LOOP A3     импульса равна 0.4 мс
    MOV AH,00
    MOV AL, 08  -вывод D3
    INT 17
A4: MOV CX,690  -задержка. При CX=0690h длительность
    LOOP A4     импульса равна 0.8 мс.
    JMP A
    INT 3

```

Программа 2. Генерирование импульсов на выходе разряда D0 без использования функций прерывания BIOS INT 17h.

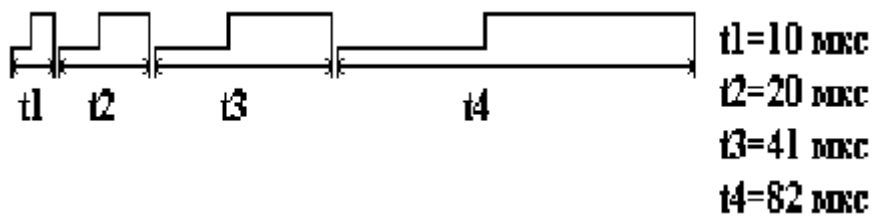
```

MOV DX,37A
MOV AL,1F
OUT DX,AL
MOV DX,378
A1: MOV AL,01
    OUT DX,AL
    MOV CX,F
A2:  LOOP A2
    MOV AL,00
    OUT DX,AL
    MOV CX,0BFF
A3:  LOOP A3
    JMP A1

```

Программа 3. Генератор импульсов на выводах D1, D2, D3, D4 частоты которых отличаются друг от друга в два раза.

управления	MOV DX,037A	- запись управляющего байта в регистр
	MOV AL,1F	
	OUT DX,AL	
	MOV DX,0378	
	MOV AL,02	- 1 во второй разряд данных
A1:	OUT DX,AL	
	MOV CX,000F	- вследствие инерционности порта в CX можно заносить число от 0 до 1FF
A2:	LOOP A2	
	INC AL	- бегущая единица по всем разрядам данных
	JMP A1	



Лабораторная работа 5
(описание)

6. « ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ IBM PC/AT » .

6.1. ЦЕЛЬ РАБОТЫ.

Целью лабораторной работы «Последовательный порт IBM PC/AT» является изучение принципов работы последовательного порта в ПЭВМ, а также приемов его программирования.

6.2 . ТЕХНИЧЕСКИЕ СРЕДСТВА.

В лабораторной работе используется: ПЭВМ IBM PC AT 386, осциллограф С1-81.

6.3 . НАЗНАЧЕНИЕ КОНТРОЛЛЕРА ПОСЛЕДОВАТЕЛЬНОГО ИНТЕРФЕЙСА.

Контроллер последовательного интерфейса предназначен для обеспечения связи по протоколу RS232C (соответствует рекомендации V.24 МККТ). Этим стандартам соответствуют ГОСТ 18145-81 и ГОСТ 23675-79 соответственно. Обычно PC имеют в своем составе два интерфейса RS-232C, которые обозначаются COM1 и COM2. Возможна установка дополнительного оборудования, которое обеспечивает функционирование в составе PC четырех, восьми и шестнадцати интерфейсов RS-232C. Интерфейс RS232C обеспечивает обмен данными между различными устройствами в последовательном формате по асинхронному методу.

Обеспечивает следующие возможности:

- 1) применение PC в качестве абонентского пункта в системах и сетях обработки данных. В этом случае PC подключается через этот интерфейс к устройствам преобразования сигналов (модемам), которые в свою очередь подключаются к каналам связи;
- 2) подключение к PC различных устройств ввода-вывода (графопостроителей, принтеров, графических манипуляторов, внешних НГМД, стримеров и т.д.);
- 3) объединение нескольких PC между собой и с другими ЭВМ для организации перекачки файлов между ними.

Широкое применение интерфейса RS-232C объясняется его универсальностью в части диапазона скоростей передачи информации (от 50 до 115 000 бит в секунду), «прозрачностью», т.е. отсутствием запрещенных к использованию для передачи данных кодовых комбинаций, наличием специализированных БИС и ИС, на которых достаточно эффективно реализуется данный интерфейс, простотой конструкции соединительных кабелей.

6.4. ПРИНЦИПЫ РАБОТЫ RS-232C.

Основные принципы обмена информацией по интерфейсу RS-232C заключаются в следующем:

обмен данными обеспечивается по двум цепям, каждая из которых является для одной из сторон передающей, а для другой приемной;

- ◆ в исходном состоянии по каждой из этих цепей передается двоичная единица, т.е. стоповая посылка. Передача стоповой посылки может выполняться сколько угодно;
- ◆ передаче каждого слова данных предшествует передача стартовой посылки, т.е. передача двоичного нуля в течение времени, равного времени передачи одного бита данных;

- ◆ после передачи стартовой посылки обеспечивается последовательная передача всех разрядов слова данных, начиная с младшего разряда. Количество разрядов слова может быть 5, 6, 7 или 8;
- ◆ после передачи последнего разряда слова данных возможна передача контрольного разряда, который дополняет сумму по модулю 2 переданных разрядов до четности или нечетности. В некоторых системах передача контрольного разряда не выполняется;
- ◆ после передачи контрольного разряда или последнего разряда слова, если формирование контрольного разряда не предусмотрено, обеспечивается передача стоповой посылки. Минимальная длительность посылки может быть равной длительности передачи одного, полутора или двух бит данных.

Обмен данными по описанным выше принципам требует предварительного согласования приемника и передатчика по количеству используемых разрядов в символе, правилам формирования контрольного разряда и длительности передачи бита данных.

Последнее согласование обеспечивается путем стандартизации ряда скоростей: 50, 75, 100, 110, 200, 300, 600, 1200, 2400, 4800, 9600, 19 200, 38 400, 57 000 или 115 000 бит в секунду. Установленная скорость должна отличаться от номинальной не более чем на 2 %, что гарантировано обеспечивается применением генераторов с кварцевыми резонаторами. Обычно используется генератор с частотой 1,8432 МГц.

Для передачи данных чаще всего используются следующие коды:

1) пятиразрядный код МТК-2 по ГОСТ 15607-84. Этот код не использует контрольный разряд. Длительность стоповой посылки равна длительности передачи полутора бит;

2) семиразрядный код КОИ-7 по ГОСТ 27463-87. Этот код использует контрольный разряд, который дополняет переданные знаки до четности. Длительность стоповой посылки равна длительности передачи одного или двух бит данных.

Выбор кода для передачи данных определяется прежде всего характеристиками подключаемого к компьютеру оборудования и обычно указывается в его документации. Скорость передачи данных также определяется подключаемым оборудованием, но, кроме того, она ограничивается еще и длиной соединительного кабеля.

6.5. ПОДГОТОВКА К ЛАБОРАТОРНОЙ РАБОТЕ.

Ознакомиться:

- ⌚ с описанием лабораторной работы «Последовательный порт IBM PC/AT»,
- ⌚ с интерфейсом RS-232C, командами отладчика AFD;
- ⌚ с программами (см. Приложение к лабораторной работе).

6.6. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

6.6.1 Написать на ассемблере программу инициализации контроллера последовательного порта (длина - 6 бит; без контроля четности; 2 стоповых бита) и цикличной передачи символа без использования прерываний.

6.6.2 Написать программу приема и передачи байта со скоростью 9600 бод с произвольными параметрами передаваемого слова, используя прерывания.

6.6.3 Написать программу приема и передачи байта со скоростью 2400 бод (длина 8 бит, с контролем по четности, 1 стоповый бит).

6.6.4 Программы отладить в среде отладчика AFD.

- 6.6.5 Зарисовать сигналы на выходе с помощью осциллографа.
 6.6.6 Оформить отчет по результатам лабораторной работы.

6.7. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.

1. Справочник «Аппаратура персональных компьютеров и ее программирование IBM PC XT/AT и PS/2». «Радио и связь», Москва, 1995г.
2. Лекции по курсу «Адаптеры и контроллеры ЭВМ».

6.8. ПРИЛОЖЕНИЕ

6.8.1 Сигналы интерфейса RS-232C

Интерфейс RS-232C имеет восемь сигналов и цепь сигнальной земли GND. В ряде случаев не все эти цепи могут быть использованы.

Наименование сигналов интерфейса RS-232C, их распределение на 9-ти и 25-ти контактных соединителях приведено в Таблице 1.

Таблица 1.

Наименование сигнала			Номер контакта соединителя	
RS-232C	МККТТ (CCITT) V.24	ГОСТ 18145-81	9-контактного	25-контактного
CF	Data carrier detect (DCD)	Цепь 109	1	8
BB	Receive Data (RX)	Цепь 104	2	3
BA	Transmit Data (TX)	Цепь 103	3	2
CD	Data Terminal Ready (DTR)	Цепь 108	4	20
CC	Data Set Ready (DSR)	Цепь 107	6	6
CA	Request to send (RTS)	Цепь 105	7	4
CB	Clear to send (CTS)	Цепь 106	8	5
CE	Ring indicator (RI)	Цепь 125	9	22
GND	GND	Цепь 102	5	7

При анализе сигналов интерфейса RS-232C необходимо иметь в виду, что этот интерфейс был задуман для сопряжения оборудования обработки данных ООД (DTE) и аппаратуры передачи данных АПД (DCE), т.е. модемов. В этом случае наименование сигналов совпадает с их функциональным назначением.

При использовании интерфейса RS-232C для подключения устройств ввода-вывода и других целей функциональное назначение сигналов может не всегда совпадать с их наименованиями.

Для определения состояния сигналов интерфейса RS-232C необходимо учитывать следующее:

■ передатчики интерфейса RS-232C вырабатывают двуполярные сигналы. Абсолютное значение амплитуды передаваемого сигнала - от 8 до 10 В. Приемники интерфейса RS-232C обеспечивают прием двуполярных сигналов с амплитудой по абсолютному значению от 3 до 10 В;

■ сигналы BB (цепь 104) и BA (цепь 103) при передаче двоичной единицы или стартовой посылки принимают значение меньше, чем минус 3 В, а при передаче двоичного нуля или стоповой посылки принимают значение больше, чем 3 В;

■ сигналы CF (цепь 109), CD (цепь 108), CC (цепь 107), CA (цепь 105), CB (цепь 106) и CE (цепь 125) в состоянии «Включено» принимают значение большее, чем 3 В, а в состоянии «Выключено» - меньше, чем минус 3 В.

6.8.2 Назначение сигналов интерфейса RS-232C

Алгоритм взаимодействия сигналов в интерфейсе RS-232C существенно зависит от типа подключаемого оборудования, поэтому ниже описано только назначение отдельных сигналов :

- ВА (цепь 103) обеспечивает передачу данных от РС;
- сигнал ВВ (цепь 104) обеспечивает прием данных в РС;
- состояние «Включено» CD (цепь 108) указывает подключаемому оборудованию, что РС готов к работе;
- состояние «Включено» сигнала CC (цепь 107) указывает РС , что подключаемое оборудование готово к работе;
- состояние «Включено» сигнала CA (цепь 105) указывает подключаемому оборудованию, что РС запрашивает разрешение на передачу данных;
- состояние «Включено» сигнала CB (цепь 106) указывает РС , что он может передавать данные сигналами ВА;
- состояние «Включено» сигнала CF (цепь 109) указывает РС, что сигналы, поступающие на вход модема из канала связи, имеют необходимый уровень;
- состояние «Включено» сигнала CE (цепь 125) указывает компьютеру, что из канала связи в модем поступает входящий вызов (на телефонном аппарате, подключенном к данному модему, должен быть звонок).

Оборудование интерфейса RS-232C не накладывает никаких ограничений на состояние сигналов интерфейса. Это значит, что управляющая программа может использовать любые комбинации из перечисленных выше сигналов для организации обмена данными.

В процессе эксплуатации RS-232C необходимо выполнять следующие требования:

- компьютер и подключенное к нему оборудование должны быть надежно заземлены через кабели электропитания, а также дополнительными проводниками, если это возможно;
- никакие кабели не должны подключаться и отключаться, пока не будет выключено электропитание всех устройств, подключенных к интерфейсу RS-232C.

6.8.3 Подготовка интерфейса RS-232C к работе

При подготовке оборудования интерфейса RS-232C к работе необходимо учитывать, что адреса портов ввода-вывода интерфейсов в системной шине РС и соответствующие им сигналы запроса прерываний IRQ изменяются механическими переключателями. Расположение этих переключателей на платах РС и их конкретное назначение должно быть указано в эксплуатационной документации или сообщаться организацией, поставляющей РС или отдельные адаптеры RS-232C.

С помощью данных переключателей возможны установки следующих режимов:

- ☐ блокирование работы каждого асинхронного интерфейса COM1 и COM2;

- ☐ блокирование формирования сигналов прерывания IRQ3 и IRQ4;
- ☐ переключение работы первого асинхронного интерфейса для работы в качестве второго.

Существуют варианты исполнения адаптеров интерфейсов RS-232C, где обеспечивается использование адресов и прерываний согласно Таблице 2.

Таблица 2.

Плата адаптера	Обозначение интерфейса RS-232C	Адрес регистра	Сигнал прерывания
Первая	COM1	3F8	IRQ4
Первая	COM3	3E8	IRQ5
Вторая	COM2	2F8	IRQ3
Вторая	COM4	2E8	IRQ5

6.8.4 Принципы построения универсального асинхронного приемопередатчика (УАПП)

Основные функции, выполняемые оборудованием интерфейса RS-232C:

- ◆ обеспечение преобразования параллельного кода в последовательный при передаче данных и обратное преобразование при их приеме;
- ◆ формирование стартового, контрольного и стопового разрядов при передаче данных;
- ◆ контроль правильности приема стартового, контрольного и стопового разрядов при приеме. Анализ их «переполнения», когда ранее принятые знаки не переписываются в системную шину до приема очередного знака;
- ◆ прием и передача знака на фиксированных скоростях;
- ◆ формирование и контроль состояния сигналов в интерфейсе RS-232C;
- ◆ организация диагностической проверки без использования подключаемого к интерфейсу RS-232C оборудования.

Оборудование, выполняющее перечисленные выше функции, реализуется в виде БИС, которые получили наименование «универсальный асинхронный приемопередатчик» УАПП (UART). УАПП реализованы в виде целого ряда БИС различных конструкций и компоновки. Эти БИС могут располагаться как на базовой плате PC, так и входить в различных комбинациях в адаптеры интерфейсов. Существуют БИС, реализующие два интерфейса RS-232C и стык Centronics. Возможны и другие варианты реализации оборудования RS-232C, но структура УАПП (назначение, адресация и логические связи входящих в их состав регистров) унифицирована и не зависит от конкретной реализации БИС.

Структурная схема УАПП приведена на рис. 1.

В состав УАПП входит 10 адресуемых регистров, а также регистр сдвига передачи, регистр сдвига приема и делитель на 16.

Для адресации регистров УАПП предусмотрено только три адресных разряда, поэтому адресация некоторых регистров обеспечивается дополнительными условиями.

Адресация регистров УАПП.

Адреса регистров УАПП портов COM1 и COM2 представлена в Таблице 3.

Таблица 3.

Код адреса регистра		Наименование регистра	Содержимое 7-го разряда (DLAB) регистра управления линией (LCR)	Сигнал	
COM1	COM2			Чтение (IOR)	Запись (IOW)
3F8h	2F8h	Буфер приема (RBR)	0	+	-
3F8h	2F8h	Буфер передачи (THR)	0	-	+
3F8h	2F8h	Делитель частоты 1 (младший байт) (DL LSB)	1	+	+
3F9h	2F9h	Делитель частоты 2 (старший байт) (DL MSB)	1	+	+
3F9h	2F9h	Регистр разрешения прерываний (IER)	0	+	+
3FAh	2FAh	Идентификатор прерывания (IIR)	0	+	-
3FBh	2FBh	Управление линией (LCR)	0	+	+
3FCh	2FCh	Управление модемом (MCR)	0	+	+
3FDh	2FDh	Состояние линии (LSR)	0	+	-
3FEh	2FEh	Состояние модема (MSR)	0	+	-
3FFh	2FFh	Неиспользуемый регистр (SR)	0	+	+

Для обеспечения передачи данных достаточно записать по адресу 3F8h в буфер передачи байт данных. После передачи данного байта в системную шину будет выдано прерывание IRQ.

Прием данных обеспечивается чтением буфера приема по адресу 3F8h. В процессе обмена данными возможно возникновение сбойных ситуаций, которые фиксируются в регистре идентификации прерываний 3FAh. При возникновении данных ситуаций вырабатывается сигнал прерывания.

УАПП обеспечивают дуплексный обмен данными, т.е. можно организовать их одновременный прием и передачу. Но все параметры обмена (скорость, формат знака и т.д.) для приема и передачи должны быть идентичны.

6.8.5 Инициализация УАПП

В соответствии с приведенной выше структурной схемой УАПП реализует следующий алгоритм инициализации:

- ✓ запись в регистр 3FBh управляющего байта с единицей в седьмом разряде;
- ✓ запись константы деления в регистры 3F8h и 3F9h;
- ✓ запись управляющего байта с нулем в седьмом разряде. Назначение остальных разрядов регистра 3FBh указано ниже;
- ✓ запись управляющего байта в регистр 3F9h;
- ✓ запись управляющего байта в регистр 3FCh.

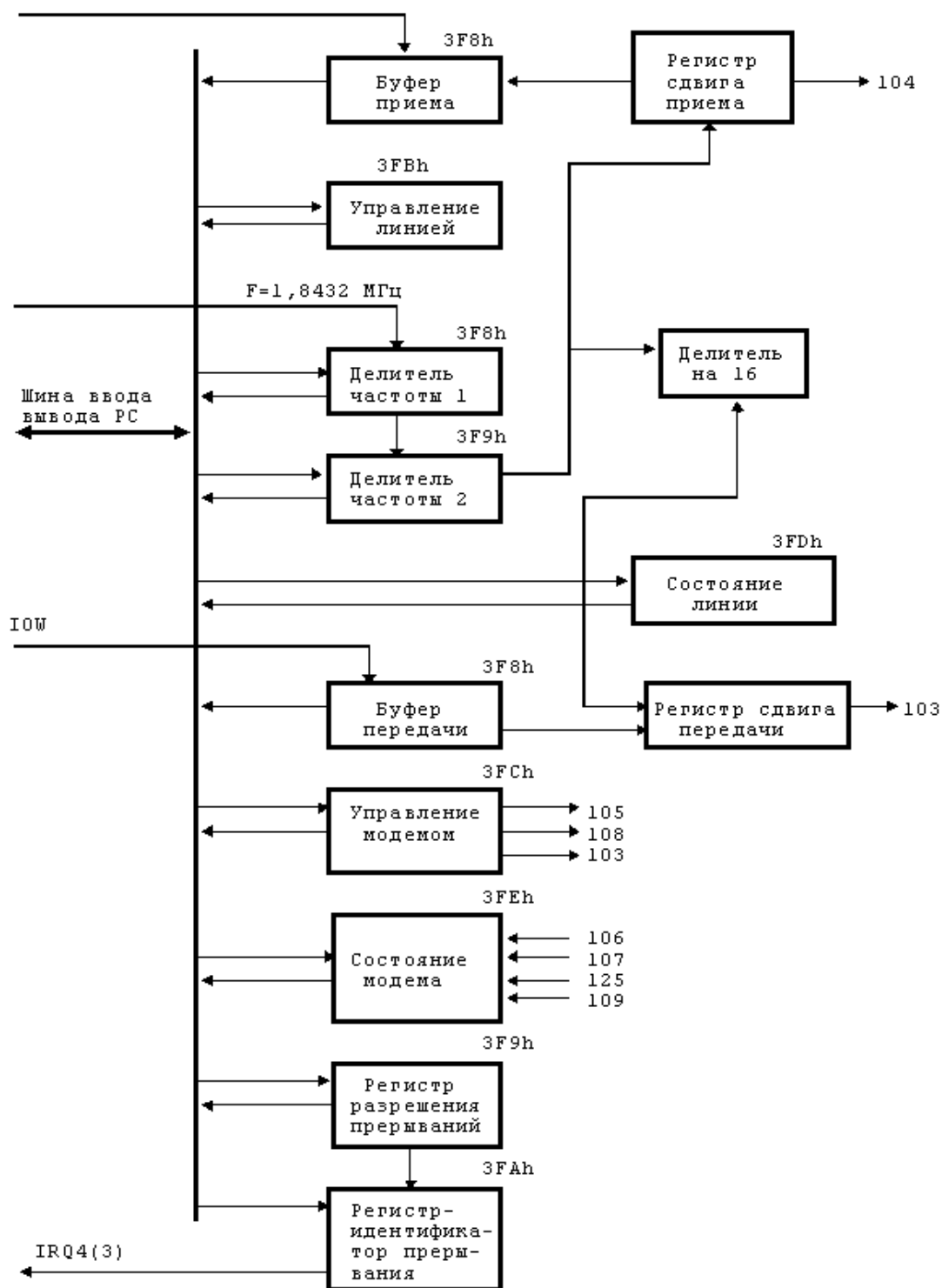


Рис.1. Структурная схема УАПП.

6.8.6 Структура регистров УАПП

Буферный регистр передачи (THR) обеспечивает промежуточное хранение передаваемых знаков данных до их записи в регистр сдвига передачи. Данный регистр доступен только по записи и при значении бита разрешения доступа к делителю (DLAB) в

регистре управления линией (LCR), равном 0. Регистр содержит восемь битов данных (бит 0 является младшим разрядом и первым посылается в канал передачи.)

Буферный регистр приема (RBR) обеспечивает промежуточное хранение принимаемых знаков данных до их чтения в системную шину PC. Этот регистр доступен только по чтению и при значении бита разрешения доступа к делителю (DLAB) в регистре управления линией (LCR), равном 0. Регистр содержит восемь битов данных (бит 0 является младшим разрядом и первым посылается в канал передачи.)

Делители частоты выбора скорости 1 и 2 служат для хранения 16-разрядной константы, которая изменяет коэффициент деления тактовой частоты на входе УАПП, обеспечивая тем самым требуемую скорость обмена данными по стыку C2. Оба регистра доступны по чтению и записи только при значении бита разрешения доступа к делителю (DLAB) в регистре управления линией (LCR), равном 1. При записи в любой из этих регистров нового значения делитель немедленно перезагружается. Частота определяется как отношение частоты задающего генератора к делителю частоты. Частота задающего генератора равна 1.8432 МГц. Соответствие между константой, загружаемой в делители частоты, и номинальной скоростью обмена данными приведено в Таблице 4.

Примечание 1. В техническом руководстве рекомендуется не устанавливать частоту свыше 19200 бод.

**Соответствие между константой и номинальной
скоростью обмена данными**

Таблица 4.

Шестнадцатеричный код константы	Номинальная скорость обмена (бит в секунду)
0900h	50
0480h	100
0240h	200
0180h	300
00C0h	600
0060h	1200
0030h	2400
0018h	4800
0010h	7200
000Ch	9600
0006h	19200
0003h	38400
0002h	57600
0001h	115200

Примечание 2. Младшие разряды константы загружаются в регистр 3F8h, старшие разряды константы в регистр 3F9h.

Регистр идентификатора прерываний (IR) хранит код условия прерывания и признак запроса прерывания. Регистр доступен только по чтению и позволяет получить информацию от контроллера о ждущем прерывании. Регистр имеет восемь разрядов (D3-D7 не используются и заполнены нулями). Формат регистра идентификатора прерываний приведен на рис. 2.

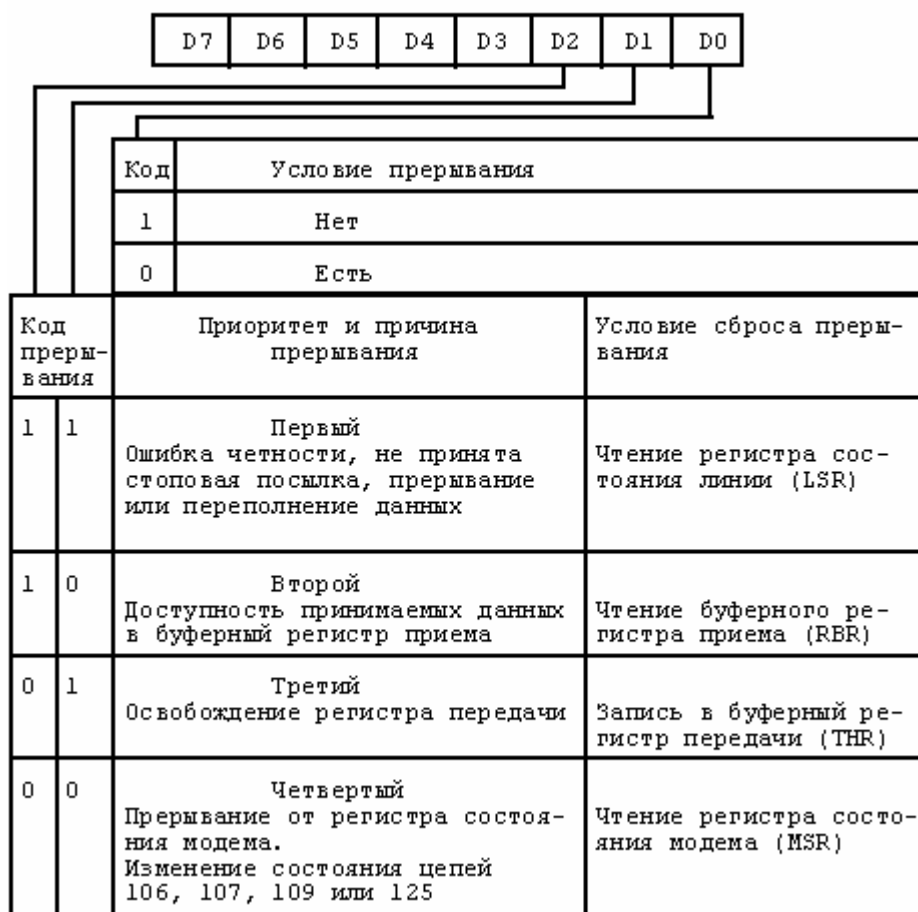


Рис. 2. Формат регистра идентификатора прерываний (IIR) 3FAh (2FAh)

Биты D2 и D1 определяют тип прерывания:

- ◆ 11 - изменилось состояние линии приемника
- ◆ 10 - принимаемые данные доступны
- ◆ 01 - освобожден регистр буфера
- ◆ 00 - изменилось состояние модема

Регистр разрешения прерываний (IER) управляет формированием сигнала запроса прерывания в соответствии с состоянием регистра. Доступен по чтению и записи, но только при значении бита разрешения доступа к делителю (DLAB) в регистре управления линией (LCR), равном 0. Регистр позволяет управлять четырьмя типами прерываний, порождаемыми контроллером последовательного интерфейса. Регистр имеет восемь разрядов (D4-D7 не используются и заполнены нулями). Формат регистра разрешения прерываний приведен на рис. 3.

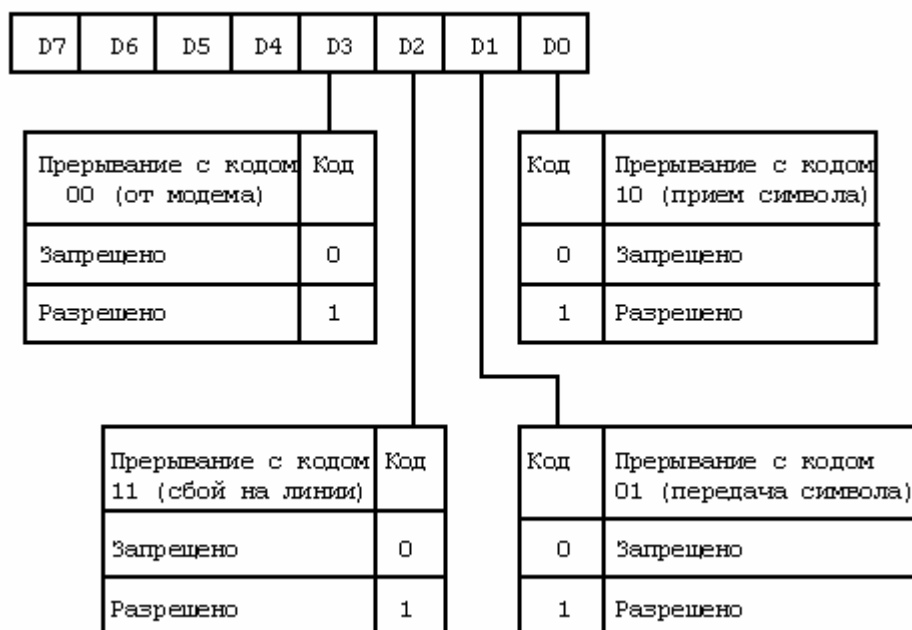


Рис. 3. Формат регистра разрешения прерываний (IER) 3F9h (2F9)

Регистр управления линией (LCR) обеспечивает выбор формата символа данных: количество информационных разрядов, количество стоповых разрядов, метод контроля и выбор адресуемых регистров УАПП. Доступен по чтению и записи. Регистр имеет восемь разрядов. Формат регистра управления линией приведен на рис. 4.

В этом режиме передаваемые данные сразу же принимаются. При этом полностью обеспечиваются прерывания приемника и передатчика.

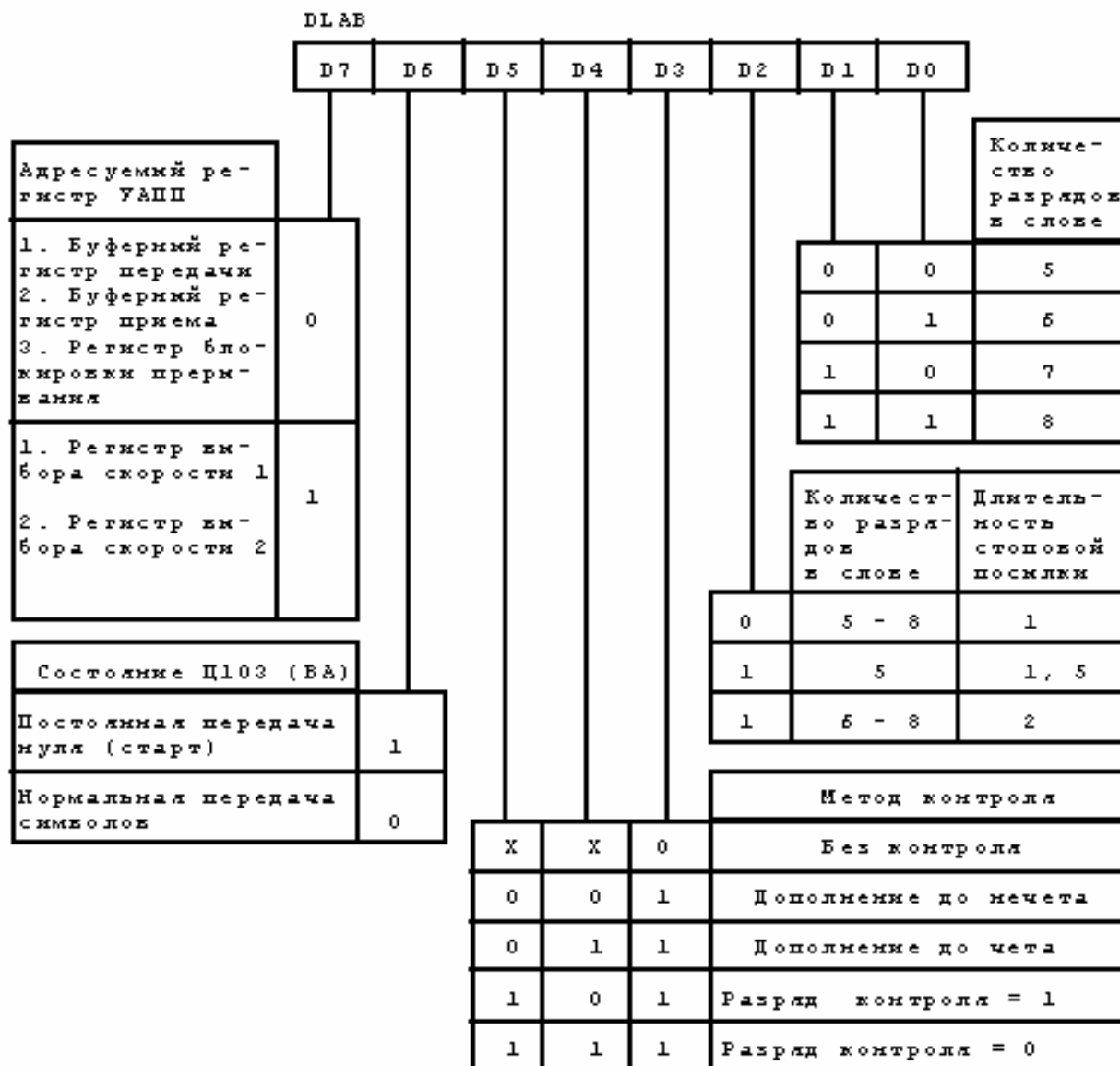


Рис. 4. Формат регистра управления линией (LCR) 3FBh (2FBh)

Регистр управления модемом (MCR) обеспечивает формирование выходных сигналов интерфейса RS-232C. Доступен по чтению и записи. Регистр имеет восемь разрядов (D5-D7 не используются и заполнены нулями). Формат регистра управления модемом приведен на рис. 5.

В диагностическом режиме УАПП обеспечивает следующие функции:

- установку в 1 сигнала DF (цепь 103);
- прекращение анализа сигналов CF (цепь 109), CC (цепь 107), CB (цепь 106), CE (цепь 125);
- прекращение анализа сигналов BB (цепь 104);
- подключение входа приемного регистра сдвига к выходу передающего регистра сдвига.

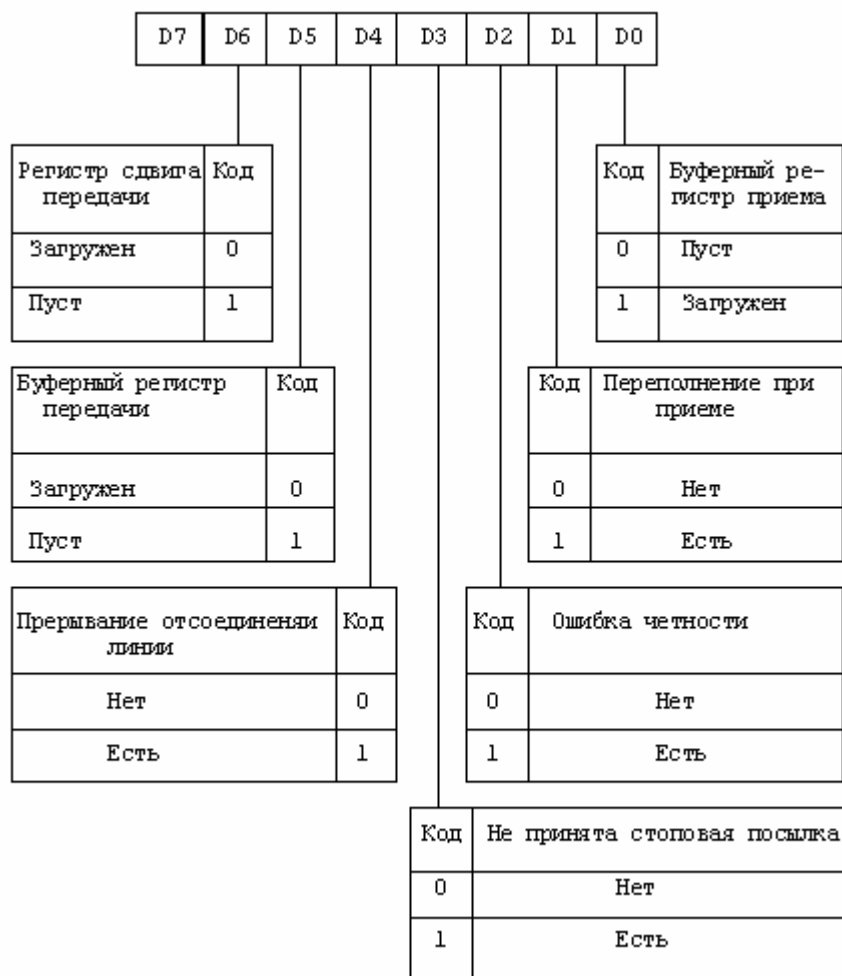


Рис. 5. Формат регистра управления модемом (MCR) 3FCh (2FCh)

Регистр состояния линии (LSR) обеспечивает хранение информации о состоянии процесса приема или передачи данных. Доступен только по чтению. Регистр имеет восемь разрядов (D7 не используется и заполнен нулем). Формат регистра состояния линии приведен на рис. 6.

Прерывание отсоединения линии (Break interrupt) указывает передающей стороне о необходимости немедленно прекратить передачу данных. Запрос на это прерывание формируется, если от принимающей станции поступает сигнал низкого уровня (нуля) в течение времени, превышающего суммарное время приема стартового, информационных, паритета и стоповых разрядов.

Состояние «Не принята стоповая посылка» означает, что стоповый разряд, который следует за последним разрядом паритета или разрядом данных, равен нулю.

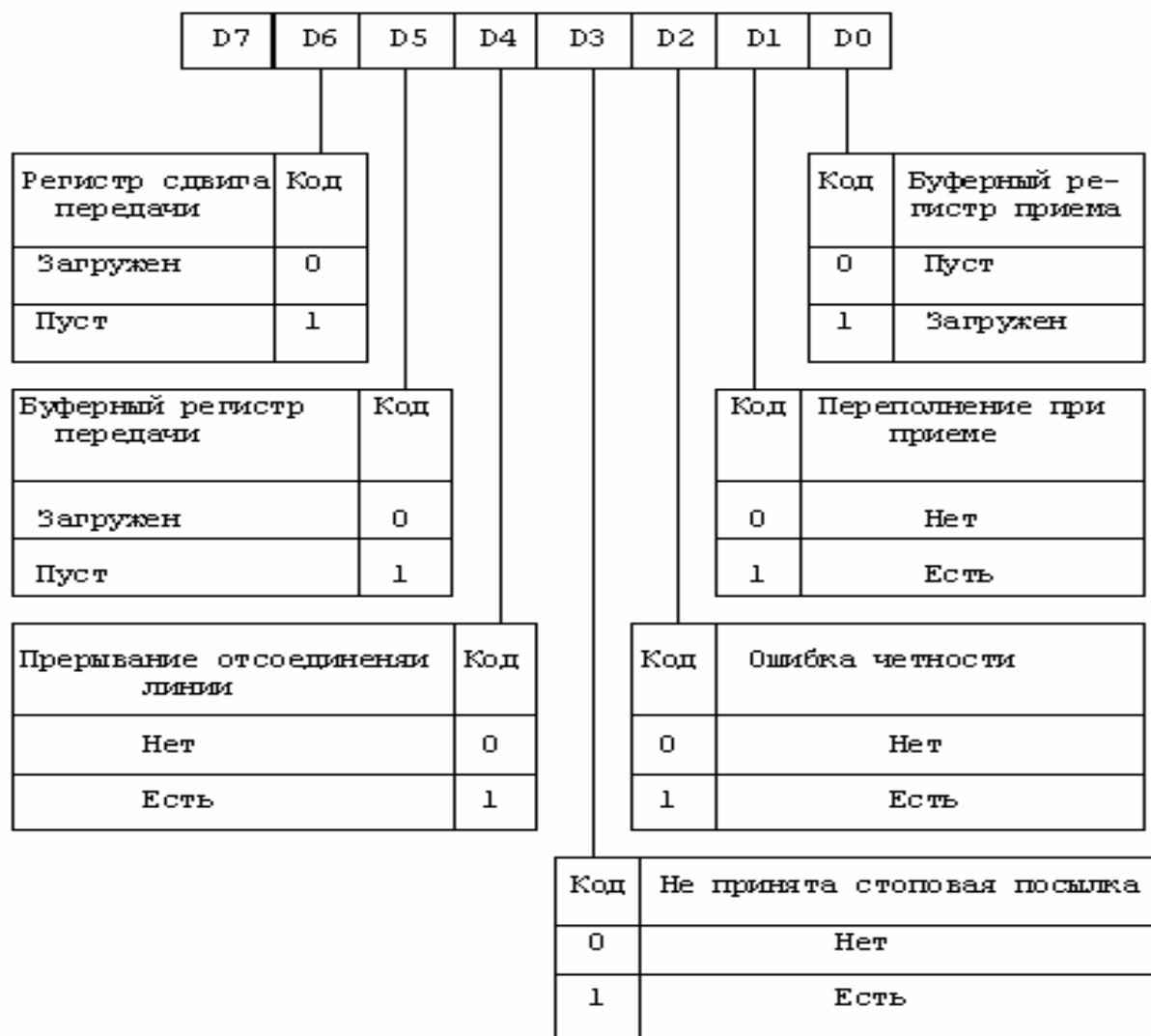


Рис. 6. Формат регистра состояния линии (LSR) 3FDh (2FDh)

Переполнение при приеме возникает, если символ данных принят в буферный регистр до того, как предыдущий символ считан из буфера. Это приводит к разрушению предыдущего символа.

Ошибка четности возникает, если в принятом символе обнаружено некорректное значение бита контроля четности.

Регистр состояния модема (MSR) обеспечивает хранение информации о управляющих линиях интерфейса RS-232C и имитацию этих сигналов в диагностическом режиме. Доступен только по чтению. Регистр имеет восемь разрядов. Формат регистра состояния модема приведен на рис. 7.

Разряды регистра состояния модема содержат информацию о приемных цепях интерфейса RS-232C и изменении их состояния. Слежение за изменением состояния позволяет упростить программу управления стыком. Особенностью контроллера 8250, представляющего собой ядро адаптера последовательного интерфейса, является возможность диагностики протокола интерфейса без подключения к линии. Реализуется управление диагностическим режимом при помощи разряда 4 регистра 3FCh (2FCh).

В диагностическом режиме значение разрядов регистра состояния модема определяется состоянием разрядов 0-3 регистра 3FCh (2FCh).

Запись в регистр возможна только в целях диагностики. Разряды D0-D3 отражают изменение состояния цепей CB-CF с момента последнего обращения к регистру.

Неиспользуемый регистр (SR) доступен по чтению и записи. Регистр не управляет контроллером и может быть использован в качестве рабочего регистра для хранения каких-либо данных.

Цепь стыка RS-232C	Изменение состояния сигналов RS-232C после чтения регистра состояния модема	Код		
CB (Ц106)	Нет	0		D0
	Есть	1		
CC (Ц107)	Нет	0		D1
	Есть	1		
CE (Ц125)	Нет	0		D2
	Есть	1		
CF (Ц109)	Нет	0		D3
	Есть	1		

Состояние приемных цепей RS-232C				Код		
Рабочий режим 3FC[4]=0		Диагностический режим 3FC[4]=1				
CB (Ц106)	Выключена	3FC[D1]	0	0		D4
	Включена		1	1		
CC (Ц107)	Выключена	3FC[D0]	0	0		D5
	Включена		1	1		
CE (Ц125)	Выключена	3FC[D3]	0	0		D6
	Включена		1	1		
CF (Ц109)	Выключена	3FC[D2]	0	0		D7
	Включена		1	1		

Рис. 7. Формат регистра состояния модема (MSR) 3FEh (2FEh)

6.8.7 Прерывания RS232

Прерывания для RS232 обеспечивают упрощенный интерфейс обращения к асинхронным портам. Предусмотрены функции для инициализации RS232, передачи символов, приема символов и чтения состояния.

Системная BIOS не поддерживает аппаратные прерывания RS232, хотя имеются два вектора прерываний: IRQ4 (INT 0Ch) для RS232-1 и IRQ3 (INT 0Bh) для RS232-2. При выполнении POST системная BIOS определяет наличие RS232 по двум стандартным базовым адресам 3F8h и 2F8h. После того как найден модуль RS232, его базовый адрес помещается в область данных BIOS, начиная с 0040:0000h. Следовательно, при программировании RS232 необходимо использовать не жестко привязанный адрес, а адрес, извлеченный из области данных BIOS.

BIOS предусматривает функцию тайм-аута для RS232, длительность которого программируется пользователем. Для задания значения тайм-аута каждому RS232 в области данных BIOS предусмотрена 4-байтовая таблица, начиная с адреса 0040:007Ch. Значения этих тайм-аутов инициализируются на 1 при включении питания и могут быть установлены до 255. Каждая единица соответствует приблизительно 900 мс.

Прерывания RS232, обслуживаемые BIOS, приведены в таблице 5.

Таблица 5.

Прерывания	Тип	Функция
INT 0Bh	HW	IRQ3, RS232-2
INT 0Ch	HW	IRQ4, RS232-1
INT 14h	SW	RS232

Когда имеет место одно из прерываний RS232, система переходит к адресу, указанному вектором для этого прерывания. В таблице 6 приведены адреса ячеек памяти, в которых помещаются эти векторы.

Драйвер RS232 (INT 14h).

INT 14h вызывается для выполнения всех функций, относящихся к вводу-выводу RS232.

BIOS: указывает на точку входа в ПЗУ драйвера RS232.

ДЕЙСТВИЕ: после входа управление передается одной из трех программ в соответствии с кодом функции в регистре AH.

Ячейки памяти, используемые для прерываний RS232

Таблица 6.

Ячейка памяти	Байт	Функция
0000:002C	4	Вектор INT 0Bh
0000:0030	4	Вектор INT 0Ch
0000:0050	4	Вектор INT 14h
0040:0000	2	Адрес базового порта RS232-1
0040:0002	2	Адрес базового порта RS232-2
0040:0004	2	Адрес базового порта RS232-3
0040:0006	2	Адрес базового порта RS232-4
0040:0071	1	Break bit
0040:007C	1	Тайм-аут для RS232-1
0040:007D	1	Тайм-аут для RS232-2
0040:007E	1	Тайм-аут для RS232-3
0040:007F	1	Тайм-аут для RS232-4

При использовании запрещенного кода функции происходит простой возврат управления. Сохраняются все регистры, кроме тех, которые возвращают значение.

НАЗНАЧЕНИЕ: прикладная программа может использовать INT 14h для прямого выполнения функций ввода-вывода на RS232.

В таблице 7 приведен перечень функций, выполняемых по команде INT 14h.

Прерывания остаются разрешенными. Маска для прерываний интерфейса и порт управления для прерываний RS232 не изменяются.

Таблица 7.

АН	Действие
00h	инициализировать RS232
01h	передать символ
02h	принять символ
03h	определить состояние

Функция 00h Инициализировать RS232(Таблицы 8, 9).

В функции указывается длина слова, число стоповых разрядов, тип паритета и скорость передачи для заданного порта.

Хотя BIOS поддерживает четыре средства RS232, при инициализации после включения питания BIOS ищет только порты 1 и 2. Для использования портов 3 и 4 пользователь должен установить базовые адреса интерфейсного порта в ячейки памяти 0040:0004h и 0040:0006h соответственно.

ВХОДНЫЕ ПАРАМЕТРЫ:

АН = 00h,

AL - конфигурация порта,

DX - номер RS232 (0-3).

Таблица 8.

AX	АН=00h	AL=Конфигурация
BX		
CX		
DX	Номер RS232	

Таблица 9.

Разряды AL							
7	6	5	4	3	2	1	0
Скорость передачи:			Паритет:		Стопковые	длина слова:	
000 - 110 бод			00 - отсутствует		биты:	10 - 7 бит	
001 - 150 бод			01 - нечетный		0 - 1 бит	11 - 8 бит	
010 - 300 бод			10 - отсутствует		1 - 2 бита		
011 - 600 бод			11 - четный				
100 - 1200 бод							
101 - 2400 бод							
110 - 4800 бод							
111 - 9600 бод							

Функция 01h Передать символ (Таблицы 10, 11).

INT 14h, АН = 01h выводит символ на заданный RS232.

ВХОДНЫЕ ПАРАМЕТРЫ:

АН = 01h,

AL - символ, который необходимо передать (00h...FFh),

DX - номер RS232 (0-3).

Таблица 10.

AX	АН=00h	AL=Символ
BX		
CX		
DX	Номер RS232	

ВЫХОДНЫЕ ПАРАМЕТРЫ:

АН - состояние линии (см. определение для INT 14h, АН = 03h).

Таблица 11.

AX	АН=Состояние линии	AL=Символ
BX		
CX		
DX	Номер RS232	

Функция 02h Принять символ (Таблицы 12, 13).

INT 14h, АН = 02h принимает символ с заданного RS232.

ВХОДНЫЕ ПАРАМЕТРЫ:

АН = 02h,

DX - номер RS232 (0-3).

ВЫХОДНЫЕ ПАРАМЕТРЫ:

АН - состояние линии (определение см. INT 14h, АН = 03h),

AL - символ, который принят (00h - FFh).

Таблица 12.

AX	АН=02h	
BX		
CX		
DX	Номер RS232	

Таблица 13.

AX	АН=Состояние линии	AL=Символ
BX		
CX		
DX	Номер RS232	

Функция 03h Определить состояние интерфейса (Таблицы 14-17).

INT 14h, АН = 03h считывает регистры модема и состояния линии заданного RS232 и помещает эти значения в выходные регистры.

ВХОДНЫЕ ПАРАМЕТРЫ:

АН = 03h,

DX - номер RS232 (0-3).

ВЫХОДНЫЕ ПАРАМЕТРЫ:

АН - состояние линии,

AL - состояние модема.

Таблица 14.

AX	АН=03h	
BX		
CX		
DX	Номер RS232	

Таблица 15.

AX	АН=Состояние линии	AL=Состояние модема
BX		
CX		
DX	Номер RS232	

Таблица 16.

Разряды АН							
7	6	5	4	3	2	1	0
1 - таймаут	1 - передатчик пуст	1 - готов к передаче	1 - пауза	1 - ошибка стоп-разряда	1 - ошибка паритета	1 - переполнение	1 - готовы данные

Таблица 17.

Разряды AL							
7	6	5	4	3	2	1	0
1 - детектор принимаемого сигнала (цепь 109)	1 - индикатор вызова (цепь 125)	1 - аппаратура передачи данных готова (цепь 107)	1 - готов к передаче (цепь 106)	0	0	0	0

6.8.8 Основы программирования контроллера последовательного интерфейса

Программное обеспечение контроллера состоит из следующих частей:

- инициализация контроллера;
- передача одного байта в канал связи;
- прием одного байта из канала связи;
- обработка прерывания, порожденного контроллером;

Перед началом обмена данными по каналу передачи выполняется начальная инициализация контроллера последовательного интерфейса.

Инициализация контроллера

1. Определить базовый адрес контроллера

```

mov    AX, 0040          ; сегментный адрес данных BIOS
mov    ES, AX            ; загрузить сегментный регистр
mov    DX, ES:[00000h]   ; выбрать базовый адрес
                        ; контроллера

```

2. Запретить прерывания от контроллера последовательного интерфейса. Для этого достаточно записать в соответствующий бит регистра IMR (регистр маски прерываний) программируемого контроллера прерываний значение 1:

```

mov    AH, 01
shl    AH, 04            ; загрузить маску для IRQ4
cli    ; запретить прерывания
in     AL, [21]          ; считать значения регистра
                        ; IMR
or     AL, AH            ; установить в 1 бит для IRQ4
out    [21], AL          ; записать значение IMR
nop    ; задержка
nop
sti    ; разрешить прерывания

```

3. Загрузить делитель программируемого генератора. Для этого необходимо сначала установить в 1 бит DLAB регистра LCR и затем загрузить младший и старший байт делителя:

```

add    DX, 0003          ; адрес регистра LCR
mov    AL, 80            ; значение бита DLAB
out    DX, AL            ; установить регистр LCR
nop
nop
sub    DX, 0003          ; адрес LSB делителя
mov    AX, 0030          ; значение делителя для
                        ; скорости
                        ; 2400 бод
out    DX, AL            ; записать LSB делителя
nop
nop

```

```

inc    DX                      ; адрес MSB делителя
mov    AL, AH                  ; значение MSB делителя
out    DX, AL                  ; записать MSB делителя
nop
nop
dec    DX                      ; базовый адрес контроллера

```

4. Установить необходимое значение регистра LCR:

```

add    DX, 0003                ; адрес регистра LCR
mov    AL, 03                  ; значение регистра LCR:
                                ; - длина символа 8 бит
                                ; - нет контроля четности
                                ; - бит DLAB имеет значение 0
out    DX, AL                  ; - записать значение в регистр
                                ; LCR
nop
nop
sub    DX, 0003                ; - базовый адрес контроллера

```

5. Установить необходимое значение регистра MCR. В этом примере программирование последовательного интерфейса без использования прерываний:

```

add    DX, 0004                ; адрес регистра MCR
sub    AL, AL                  ; установить управляющие
                                ; сигналы
                                ; модема в 0
out    DX, AL                  ; записать значение в регистр
                                ; MCR
nop
nop
sub    DX, 0004                ; базовый адрес контроллера

```

6. Установить необходимое значение регистра IER.

```

inc    DX                      ; адрес регистра IER
sub    AL, AL                  ; контроллер не порождает
                                ; прерывания
out    DX, AL                  ; записать значение в регистр
                                ; IER
nop
nop
dec    DX                      ; базовый адрес контроллера

```

Передача одного байта в канал связи

Для передачи одного байта (символа “!” - ASCII-код 21h) в канал связи необходимо дождаться освобождения регистра THR (регистр буфера передатчика) и поместить значение передаваемого значения в этот регистр:

Для приема одного байта данных из канала связи необходимо дождаться состояния «готовность принимаемых данных» и прочитать один байт данных из регистра RBR (регистр буфера приемника):

	mov	CX, 08	; счетчик повторений
	add	DX, 0005	; адрес регистра LSR
InLSR:	in	AL, DX	; прочитать значение регистра
			; LSR
	test	AL, 1E	; одно из ошибочных
			; состояний
	jnz	A	
	test	AL, 01	; доступность принимаемых
			; данных
	jnz	InData	; да, переход
	loop	InLSR	; нет, продолжить опрос
	jmp	A	
InData:	sub	DX, 0005	; адрес регистра RBR
	in	AL, DX	; прочитать принятый символ
A:	int 3		

mov	AX,0040	;
mov	ES,AX	; определение базового адреса
mov	DX,ES:[0000]	;
mov	AH,00	
mov	AL,A3	; формат слова :
		; - 8 бит
		; - 1 стоповый бит

```

MOV    DX,0000
INT     14                                ; инициализация

MOV    CX,0FFF                            ; число повторений
A: MOV  AH,01
MOV    AL,21                              ; символ "!"
MOV    DX,0000
INT     14                                ; передача символа
LOOP   A
INT 3

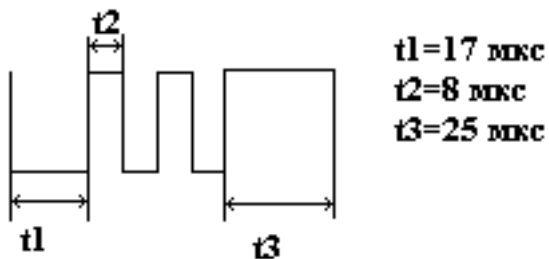
```

Программа 1. Инициализации последовательного порта (длина -6 бит, без контроля четности, 2 стоповых бита и циклической передачи символа)

```

MOV DX,3FB
MOV AL,05
OUT DX,AL
MOV DX,3F9
MOV AL,00
OUT DX,AL
MOV DX,3FD
A1: IN  AL,DX
TEST AL,40
JZ  A1
MOV DX,3F8
MOV AL,05
A2: OUT DX,AL
MOV CX,FF
LOOP A2
INT 3

```



Программа 2. Инициализация последовательного порта COM1 (длина -6 бит, без контроля по четности, 2 стоповых бита) и циклическая передача символа, скорость передачи 2400 бод.

```

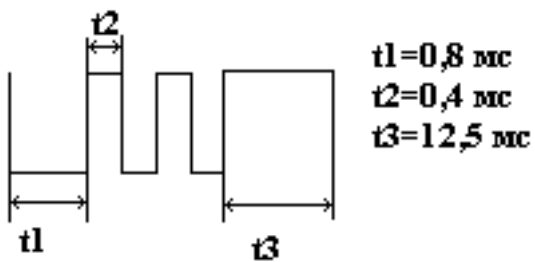
MOV DX,3FB
MOV AL,85      - установка бита DLAB
OUT DX,AL
MOV AL,30      - запись младшего байта делителя
MOV DX,3F8

```

```

OUT DX,AL
MOV AL,00          - запись старшего байта делителя
MOV DX,3F9
OUT DX,AL
MOV DX,3FB          - инициализация последовательного порта COM1 длина
-6 бит, без контроля по четности, 2 стоповых бита
MOV AL,05
OUT DX,AL
MOV DX,3F9          - запрет прерываний
MOV AL,00
OUT DX,AL
MOV DX,3FD          - чтение состояния линии
A1: IN AL,DX
TEST AL,40          - регистр передатчика пуст
JZ A1
MOV DX,3F8          - регистр данных
MOV AL,05
A2: OUT DX,AL
MOV CX,FF
LOOP A2              - циклический вывод константы 05h
INT 3

```



**Лабораторная работа 6
(описание)**

7 . «ПРОГРАММИРОВАНИЕ КЛАВИАТУРЫ

7.1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы « Программирование клавиатуры» является ознакомление с принципами работы клавиатуры и адаптера клавиатуры, изучение систем команд клавиатуры и адаптера клавиатуры.

7.2. ТЕХНИЧЕСКИЕ СРЕДСТВА , ИСПОЛЬЗУЕМЫЕ В РАБОТЕ

В лабораторной работе используется ПЭВМ IBM PC AT 386.

7.3. ОПИСАНИЕ.

7.3.1 Клавиатура.

Клавиатура - это основное средство связи пользователя с машиной. В клавиатуре имеется микропроцессор (далее МП), который сканирует ее в поиске нажатых клавиш. Этот МП контролирует также свою линию связи с контроллером клавиатуры. По этой линии связи от адаптера поступают команды управления клавиатурой, а адаптеру - коды сканирования и подтверждения. Коды сканирования клавиатуры вырабатываются при нажатии и отпуске клавиши.

Для персональных компьютеров, совместимых с IBM PC, были разработаны три типа клавиатуры:

- 1) 83-клавишная (использовалась в PC XT). Она имеет следующие группы клавиш:
 - основные клавиши (клавиатура пишущей машинки);
 - функциональные клавиши (F1-F10);
 - клавиши управления курсором (клавиатура калькулятора);
 - клавиши состояния и сдвигов (Shift, Ctrl, Alt, Caps Lock и др.).

2) 84-клавишная (клавиатура PC AT) отличается от PC XT дизайном (изменились размер и расположение некоторых клавиш). Кроме того, контроллер клавиатуры стал обладать возможностями программирования, расширенной обработки ошибок, повторяемостью клавиш.

3) 101/102-клавишная клавиатура PS/2. На ней имеется множество новых клавиш: F11, F12, правые клавиши Ctrl, Alt, вторая клавиша Enter, клавиша Pause, полностью продублированы клавиши управления курсором. Появилась возможность выбора одного из трех режимов работы (№1,2,3), устанавливаемых командой F0h. В настоящее время со всеми моделями ПК поставляется 101-клавишная клавиатура. На ПК ЕС 1840, ЕС 1841 была установлена клавиатура, функциональным аналогом которой являлась клавиатура PC XT. На программном уровне клавиатура PC AT и 101-клавишная клавиатура совместимы между собой и обладают всеми возможностями клавиатуры PC XT.

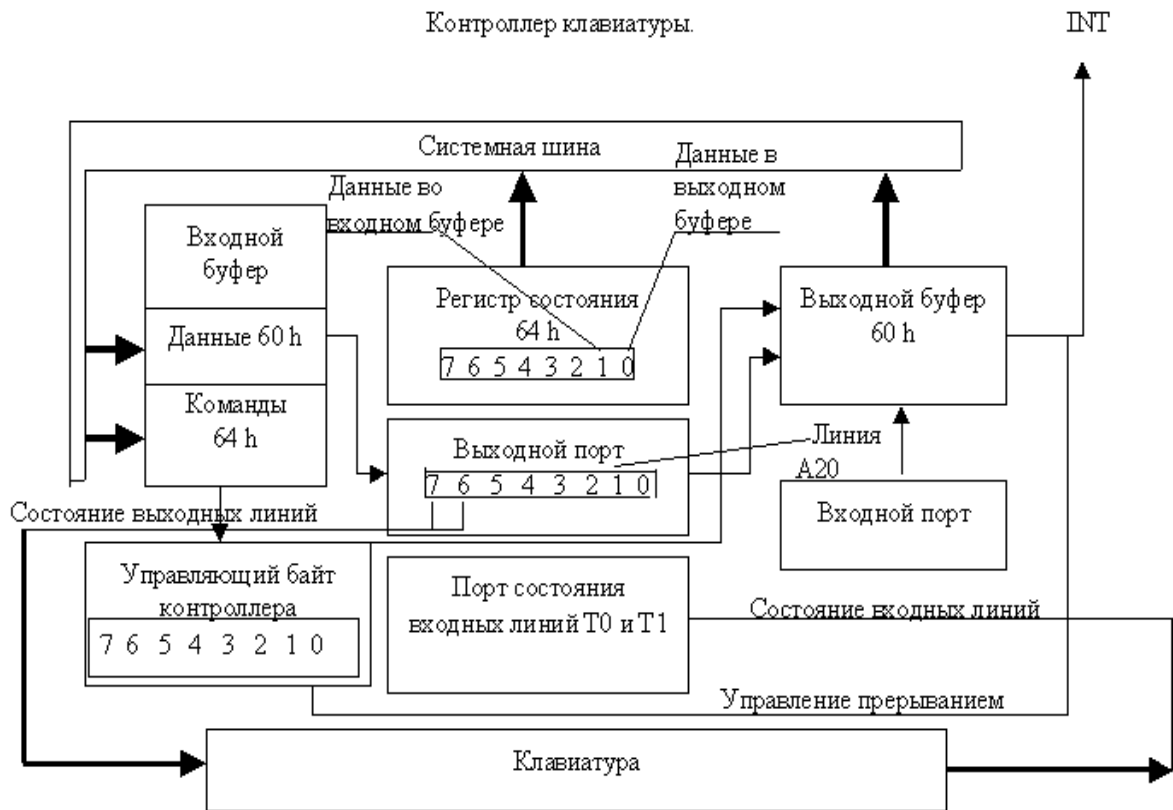
7.3.2 Функционирование адаптера клавиатуры и клавиатуры.

Адаптер клавиатуры можно рассматривать с двух точек зрения: во-первых, адаптер взаимодействует непосредственно с клавиатурой, принимая от нее информацию, и может программироваться для задания режимов такого взаимодействия, а также для задания некоторых характеристик клавиатуры; во-вторых, адаптер взаимодействует с процессором, извещая его о получении информации от клавиатуры, а также принимая управляющую информацию от программы.

Адаптер клавиатуры выполняет следующие основные

функции:

- прием данных от клавиатуры по последовательному интерфейсу;
- проверка четности поступившего байта информации;
- трансляцию скан-кодов (кодов клавиш);
- помещения байта данных в выходной буфер и извещение процессора;
- передачу байта информации из входного буфера в клавиатуру.



Рассмотрим взаимодействие адаптера с процессором и клавиатурой.

Регистр состояния - это восьмибитовый регистр, доступный процессору только по чтению по адресу 64h.

Входной буфер представляет собой два восьмибитовых регистра, доступные по записи с адресами 60h и 64h. Причем первый из регистров используется для записи данных, а второй - для записи команд.

Выходной буфер - восьмибитовый регистр, доступный только по чтению, с адресом 60h.

Входной порт содержит восьмибитовое значение, определяющее конфигурацию компьютера.

Выходной порт содержит восьмибитовое значение, определяющее режим работы компьютера, а также состояние выходных линий интерфейса с клавиатурой. Выходной порт доступен по чтению и записи посредством специальных команд контроллера клавиатуры.

Порт состояния входных линий T0 и T1 содержит состояние входных линий синхронизации и данных, которое может быть считано командой контроллера клавиатуры.

Управляющий байт адаптера клавиатуры управляет

режимами работы адаптера, значение байта может быть считано и записано посредством команд адаптера.

При передаче и приеме информации от клавиатуры используется последовательный интерфейс с длиной посылки 11 бит. Первым идет признак начала посылки, затем следуют восемь битов данных, далее бит четности и в конце признак конца посылки. Для синхронизации обмена информацией между адаптером и клавиатурой используется специальный генератор в клавиатуре. Из клавиатуры в выходной буфер адаптера передаются служебные коды и коды клавиш, а от адаптера через входной буфер в клавиатуру поступают управляющие команды. Каждая нажатая на клавиатуре клавиша вырабатывает свой код, который передается в адаптер, причем в клавиатуре устроен аппаратный 17-байтовый FIFO буфер для сохранения кодов, предназначенных для передачи в адаптер. 16 его байтов используются для сохранения собственно кодов, а в 17-ый байт записывается признак переполнения буфера (00h), в том случае, когда по заполнению буфера нажимается очередная клавиша, а адаптер клавиатуры оказывается еще не готовым к приему информации от клавиатуры. Прием данных от клавиатуры может быть заблокирован с помощью ключа (переключатель обычно располагается на передней панели системного блока). Однако, в этом случае разрешена передача и прием команд, и только коды клавиш, получаемые от клавиатуры, игнорируются адаптером. При перемещении байта информации, полученного от клавиатуры, в выходной буфер адаптер выставляет прерывание по линии IRQ1, которое обычно обрабатывается в BIOS, как INT 09h.

Регистр состояния адаптера содержит бит, который может указывать на ошибку, возникшую при приеме байта информации из клавиатуры. Причиной ошибки может быть нарушение четности или сбой синхронизации, когда передача байта оказывается незавершенной в течении заданного времени (обычно это 2 мс). В случае любой из ошибок в выходной буфер адаптера помещается значение FFh.

Передача каждого байта из клавиатуры в адаптер должна завершиться подтверждением его приема адаптером, в противном случае клавиатура блокируется: не принимает и не передает информацию. Единственным исключением является прием байта с нарушением бита четности, в таком случае адаптер автоматически посылает в клавиатуру команду повторить передачу (Resend), а клавиатура отвечает повторной передачей информации.

Передача от адаптера клавиатуры происходит аналогичным образом. Адаптер формирует 11-битовую посылку. Аналогично приему возможны два типа ошибок: нарушение четности и нарушение синхронизации. Возникновение одной из этих ошибок отражается в регистре состояния, а в выходной буфер помещается значение FEh. Для каждого передаваемого байта ожидается подтверждение от клавиатуры о его получении. Для каждой клавиши клавиатура вырабатывает два кода: код нажатия и код отжатия клавиши. Вернее, во втором случае вырабатывается тот же код, что и при нажатии клавиши, но ему предшествует посылка шестнадцатеричного значения F0h.

Адаптер клавиатуры занимается тем, что преобразует коды, получаемые от клавиатуры, в порядковые номера клавиш, обычно называемые системными скан-кодами. При этом двухбайтовый код отжатия от клавиатуры заменяется однобайтовым значением со старшим битом, установленным в единицу.

7.3.3 Программирование клавиатуры.

Как после нажатия, так и после отжатия клавиши адаптер клавиатуры генерирует аппаратное прерывание IRQ1. Обработчик прерываний INT 09h читает из порта 60h байт данных и начинает обработку нажатия (отжатия) клавиши.

Работа с клавиатурой через порт ввода/вывода.

Доступ к клавиатуре через порты ввода/вывода является самым низкоуровневым методом доступа. С помощью порта 64h можно определить доступна ли клавиатура, и разрешить (запретить) доступ к ней. Причем, если значение, возвращаемое из порта 64h, содержит в четвертом разряде ноль, то клавиатура недоступна. Порт 60h предназначен для чтения данных с клавиатуры и передачи команд и кодов в клавиатуру. С помощью порта 64h можно управлять работой адаптера клавиатуры (микроконтроллер I8042, расположенный на системной плате PC/AT) записью по этому адресу соответствующих кодов команд адаптера клавиатуры, читать состояние его портов и пр.

Команды управления клавиатурой.

Команды клавиатуры представлены в Таблице 1 а в Таблице 2 - коды особых случаев, посылаемые клавиатурой адаптеру.

Таблица 1.

Код	Функция
EDh	Установить индикаторы состояния
EEh	Эхо (средство диагностирования)
F0h	Выбрать альтернативный режим
EFh – F1h	Резерв – холостые команды (Nor)
F2h	Идентификатор расширенной клавиатуры
F3h	Установить частоту повторения клавиш
F4h	Разрешить клавиатуру
F5h	Запрет по умолчанию
F6h	Установить условия по умолчанию
F7h – FAh	Установить операции автоповтора клавиатуры
FBh – FDh	Установка автоповтора отдельных клавиш
FEh	Повтор последней передачи
FFh	Сброс и тест клавиатуры
ADh	Запрет сканирования
AEh	Разрешение сканирования

Таблица 2.

Код	Назначение
FAh	АСК – подтверждение приема команды
AAh	Внутренний тест прошел нормально
FDh	Ошибка внутреннего теста
EEh	Ответ на эхо-команду
00h – FFh	Переполнение буфера или неидентифицированная клавиша
FEh	Запрос повторной передачи

Описание команд.

1. Установка и сброс индикаторов состояния (Edh).

На клавиатуре обычно расположены по крайней мере 3 светодиодных индикатора, которые отражают состояние переключателей NUM LOCK, CAPS LOCK, SCROLL LOCK. Состоянием этих индикаторов можно управлять, выдавая в клавиатуру команду EDh. После получения этой команды клавиатура прекращает сканирование, возвращает системе EAh - код подтверждения, и ожидает от системы байта конфигурации, который определяет установку светодиодных индикаторов. После получения байта конфигурации клавиатура устанавливает светодиодный индикатор, возвращает код ACK (код EAh) и возобновляет сканирование. Если в то время, когда клавиатура ожидает байт конфигурации, поступает другая команда, выполнение EDh прекращается и выполняется новая команда.

Байт конфигурации имеет следующий формат:

разряд 0 - 1, если SCROLL LOCK включен

разряд 1 - 1, если NUM LOCK включен

разряд 2 - 1, если CAPS LOCK включен

разряды 3-7 должны быть установлены в ноль.

2. Эхо (EEh).

Данная команда используется для диагностирования. После получения этой команды клавиатура возвращает ответ - «эхо» (EEh).

3. Выбрать альтернативный режим (F0h).

Данная команда позволяет выбрать один из трех возможных режимов работы клавиатуры [1,2,3] (набор SCAN-кодов клавиатуры). В качестве параметра команды можно посылать 01, 02, 03 для выбора соответствующего набора SCAN-кодов. Клавиатура подтверждает как получение самой команды, так и получение параметра. Параметр может принимать значение 00h, тогда вслед за посылкой подтверждения приема параметра, клавиатура отвечает еще одним байтом с номером используемой таблицы SCAN-кодов (программа 3 в приложении).

4. Холостые операции (EFh-F1h).

«Холостые операции» (EFh-F1h): в ответ на такую команду клавиатура посылает код повторной посылки FEh, не предпринимая никаких других действий.

5. Идентификатор расширенной клавиатуры (F2h).

По этой команде клавиатура посылает подтверждение, прекращает сканирование, а затем посылает последовательно 2 байта идентификатора и возобновляет сканирование (программа 1 в приложении).

6. Установить частоту повторения клавиши (F3h).

Эта команда изменяет значение задержки и частоты при повторе клавиши. Как только клавиатура получает команду F3h, она прекращает сканирование и вызывает код ACK. После этого система посылает клавиатуре байт, в котором содержатся значения частоты повторения и времени задержки. Частота повторения показывает, сколько раз в секунду повторяется данная клавиша. Время задержки - это время, в течении которого клавиша должна удерживаться нажатой, прежде чем начнется функция повторения. Клавиатура возвращает код ACK и остается в предшествующем состоянии сканирования.

Величина начальной задержки D и частоты повтора (период) P в миллисекундах могут быть рассчитаны по

следующим формулам:

$$D=(1+2*B6+B5)*250$$

$$P=4.17*(8+4*B2+2*B1+B0)*2^{(2*B4+B3)},$$

где B6-B0-значения соответствующих битов байта-параметра, бит 7 при этом не используется и должен быть равен 0. Для расчетной величины задержки D допустимы отклонения до 20 %.

По умолчанию после начальной загрузки устанавливаются следующие значения:

$$D=500\pm 25 \text{ (мс)}$$

$$P=81.7\pm 16 \text{ (мс)}$$

Для возобновления сканирования должна быть передана команда F4h (разрешить клавиатуру).

Байт частоты повторения и времени задержки имеет следующий формат:

- частота повторения (число повторений в секунду) - разряды 4-0.

- время задержки (в миллисекундах) - разряды 5, 6 .

8. Разрешить клавиатуру (F4h).

Команда «Разрешить клавиатуру (F4h)» разрешает клавиатуре начать сканирование. После получения этой команды клавиатура возвращает код АСК, сбрасывает выходной буфер и начинает сканирование.

9. Блокировка клавиатуры (0F5h).

Данная команда предназначена для приведения всех характеристик клавиатуры в исходное состояние. Клавиатура подтверждает прием команды, чистит выходной буфер, устанавливает набор SCAN-кодов 3 и устанавливает значения частоты повтора клавиши, начальную задержку, принятые по умолчанию. Затем клавиатура блокируется и переходит в ожидание дальнейших инструкций.

10. Установить условия по умолчанию (F6h).

По команде «Установить условия по умолчанию (F6h)» клавиатура посылает код АСК и

сбрасывает частоту повторения клавиши и время задержки;

очищает буфер клавиатуры;

устанавливает типы клавиш по умолчанию (только для режима 3);

сбрасывает последнюю повторяемую команду;

продолжает сканирование.

11. Повторить передачу (FEh).

Команда выдается только после передачи данных с клавиатуры. Клавиатура отвечает повторной передачей последнего байта, переданного системе. Если последний переданный байт был RESEND (FEh), клавиатура передает предшествующий ему байт.

2. Сброс (FFh).

Клавиатура возвращает системе код АСК. Для того, чтобы принять этот код, система возбуждает линии CLOCK и DATA по меньшей мере на 500 мкс. После получения команды клавиатура запрещается либо до принятия системой кода АСК, либо до выдачи на клавиатуру другой команды.

7.3.4 Режимы работы клавиатуры.

У 101-клавишной клавиатуры есть три режима работы - 1, 2 и 3. Каждый из этих режимов может быть выбран с помощью команд от системы.

Режим 1. Для установки этого режима необходимо выполнить команду контроллера A0h с байтом конфигурации 01h. В наборе кодов сканирования режима 1 каждой клавише присваивается базовый код сканирования, а в некоторых случаях и дополнительные коды для генерирования в системе искусственных состояний переключения регистров. Коды сканирования автоматического повторения каждой клавиши идентичны одиночным кодам сканирования.

Режим 2. Это устанавливаемый по умолчанию режим расширенной клавиатуры, который система выбирает при инициализации после включения питания. В этом режиме код сканирования адаптер клавиатуры 8042 преобразует генерируемые клавиатурой коды нажатия в системные коды, требуемые BIOS. За исключением системных кодов новых клавиш расширенной клавиатуры системные коды соответствуют кодам сканирования, которые генерируются клавиатурой в режиме 1.

Режим 3. В этом режиме должно быть запрещено преобразование кодов у адаптера клавиатуры 8042, поскольку адаптер не может преобразовать этот набор кодов сканирования. В наборе кодов сканирования 3 каждой клавише присваивается уникальный 8-разрядный код нажатия, который передается при нажатии клавиши. При отжатии каждая клавиша отправляет код отжатия. Код отжатия состоит из двух байтов, первый из которых является префиксом отжатия (F0h), а второй идентичен коду нажатия данной клавиши. В этом наборе кодов каждая клавиша посылает только один код сканирования и состояние любой клавиши не зависит от состояния других.

7.3.5 Регистры и порты адаптера клавиатуры.

Регистр состояния адаптера клавиатуры доступен по чтению по шестнадцатеричному адресу 64 h. Биты регистра несут информацию о состоянии адаптера и интерфейса с клавиатурой. Регистр состояния может быть считан в любой момент времени. Формат регистра состояния приведен в Таблице 3.

Таблица 3.

7	6	5	4	3	2	1	0
PE	RTO	TTO	IS	CD	SF	IBF	OVF

PE отражает состояние ошибки четности при приеме байта от клавиатуры. Если значение бита равно 1, то последний принятый байт содержал четное число единиц, что является ошибкой.

RTO устанавливается в 1, если прием байта от клавиатуры не был закончен в заданный интервал времени, что является ошибкой.

TTO устанавливается в 1, если передача байта в клавиатуру не была завершена. Если причина ошибки при передаче-отсутствие тактирования со стороны клавиатуры, то только этот бит будет установлен в 1. Если клавиатура не ответит на переданный байт в заданный интервал времени, то будут установлены в 1 два бита RTO и TTO. Если в ответ на переданный в клавиатуру байт будет получен байт с нарушением четности, то будут установлены в 1 два бита PE и TTO.

IS отражает состояние переключателя блокировки клавиатуры. Бит установлен в 0, если клавиатура заблокирована.

CD указывает адаптеру клавиатуры на то, как

интерпретировать байт, помещенный во входной буфер. Если запись производилась по адресу 60h, то записанный байт считается данными и бит CD устанавливается в 1. Если запись производилась по адресу 64 h, то бит CD устанавливается в 0 и байт во входном буфере рассматривается как команда.

SF принимает значение 0 после включения питания компьютера, после завершения начального тестирования бит устанавливается в 1.

IBF отражает состояние входного буфера адаптера клавиатуры (шестнадцатеричные значения 60 h и 64 h). Данный бит равен 0, если буфер пуст. При записи во входной буфер байта информации, который адаптер еще не готов считать для передачи в клавиатуру, бит IBF устанавливается в 1.

OVF отражает состояние выходного буфера адаптера клавиатуры (шестнадцатеричный адрес 60 h). Данный бит равен 0, если выходной буфер пуст. При записи в выходной буфер байта от клавиатуры адаптер устанавливает бит OVF в 1. После считывания системой байта из выходного буфера бит снова устанавливается в 0.

Порты адаптера клавиатуры не доступны процессору напрямую, т.е. он не может их адресовать как регистр состояния и входные или выходные буферы. Вместо этого адаптер предоставляет специальные команды для чтения данных входного и выходного портов, а также для записи значения выходного порта (команды C0 h, D0 h и D1 h соответственно).

Ниже приведены значения битов, определенные для данных портов адаптера клавиатуры.

Входной порт адаптера клавиатуры определяет некоторые параметры конфигурации компьютера, как это показано в Табл. 4.

Таблица 4.

7	6	5	4	3	2	1	0
В 7	В 6	В 5	В 4	Р е з е р в			

В7 указывает, заблокирована ли клавиатура ключом (0-заблокирована, 1-не заблокирована).

В6 определяет тип видеоадаптера (0-цветной графический адаптер, 1-монохромный адаптер).

В5 задает значение переключателя фирмы-изготовителя (0-переключатель установлен, 1- переключатель не установлен).

В4 указывает количество RAM на системной плате (0-не использовать вторые 256 К памяти, 1- использовать вторые 256 К памяти).

Биты 3-0 не определены.

Следует отметить, что состояние входного порта может быть разным в разных компьютерах и, вообще говоря, не несет в себе ценной для программиста информации. Скорее всего наличие входного порта является атавизмом, оставшимся от самой первой модели IBM PC/AT, поэтому не рекомендуется работать с этим портом и полагаться на его значение.

Выходной порт адаптера клавиатуры отражает состояние сигналов на некоторых выходных линиях. При этом значения битов 1 и 0 отражают соответственно нижний и верхний уровень сигнала на линии, как это показано в Таблице 5.

Таблица 5.

7	6	5	4	3	2	1	0
В 7	В 6	В 5	В 4	Р е з е р в		В 1	В 0

Бит В7 определяет значение линии данных при передаче их в клавиатуру.

Бит В6 определяет значение линии синхронизации при передаче данных в клавиатуру.

Бит В5 указывает, пуст ли входной буфер, бит устанавливается в 1, если входной буфер пуст.

Бит В4 содержит значение 1, если выходной буфер полон, и значение 0 в противном случае.

Содержимое битов 3 и 2 не определено.

Бит В1 определяет состояние дополнительной адресной линии А20.

Бит В0 подсоединен к линии сброса системы (System Reset). Переустановка значения этого бита в 0 приводит к сбросу системы.

Порт состояния входных линий Т0 и Т1.

Помимо описанных портов адаптера клавиатуры существует порт состояния входных линий Т0 и Т1. Его значение также может быть считано только специальной командой контроллера (Е0h). Значение порта определено в Табл. 6.

Таблица 6.

7	6	5	4	3	2	1	0
Р е з е р в						Т 1	Т 0

Биты 7-2 не определены.

Биты Т0 и Т0 определяют значения входной линии данных и входной линии синхронизации соответственно.

7.3.6 Команды адаптера клавиатуры.

Команды адаптера клавиатуры представляют собой набор кодов, посылаемых процессором во входной буфер команд с шестнадцатеричным адресом 64h. В том случае, если у команды существует параметр, он записывается сразу после записи кода команды во входной буфер данных (шестнадцатеричный адрес порта 60 h).

Некоторые коды команд адаптера (выполненного на базе микроконтроллера 8042) представлены в Таблице 7.

Таблица 7.

Код	Функция
20h	Прочитать в порт 60h текущий командный байт
60h	Загрузить новый командный байт
A5h	Специальное чтение
AAh	Инициализация
ABh	Тест интерфейса
ACh	Диагностический дамп
ADh	Блокирование клавиатуры
AEnh	Разблокирование клавиатуры
C0h	Прочитать входной порт
D0h	Прочитать выходной порт
D1h	Записать в выходной порт
E0h	Прочитать входы TEST0 и TEST1
F0h - FFh	Пульсация выходного порта

1). Считать (командный) управляющий байт (20h).

Код 20h посылается по адресу 64h. Адаптер в ответ помещает в порт 60h значение текущего управляющего байта. Он может быть прочитан процессором командой IN с адресом 60h.

2). Загрузить новый командный байт (60h).

Это 2-байтовая операция. Чтобы записать новый командный байт, необходимо

1) записать 60h в порт 064h,

2) записать новый командный байт в порт 060h

Формат командного байта адаптера представлен в Таблице 8.

Таблица 8.

7	6	5	4	3	2	1	0
0	IBM	DAD	DK	DL	SF	EAI	EKI

IBM - режим совместимости с IBM PC. Запись единицы в этот разряд программирует адаптер на конвертирование кодов сканирования, полученных с клавиатуры, по принципам IBM PC. Это включает в себя конвертирование двухбайтных последовательностей кодов отжатия в однобайтный формат IBM PC.

0 - не преобразовывать коды клавиатуры,

1 - преобразовывать коды клавиатуры в коды сканирования

8088/8086.

DAD - режим IBM PC. Запись единицы в этот разряд программирует адаптер на поддержку интерфейса IBM PC. В этом режиме контроллер не контролирует паритет и не конвертирует коды сканирования.

0 - использовать 11-разрядные коды клавиатуры (PC AT/80386/80286),

1 - использовать коды клавиатуры, совместимые с PC XT/8088/8086

DK - заблокировать клавиатуру: 0 - разрешить интерфейс клавиатуры; 1 - запретить интерфейс клавиатуры

DL - отмена блокировки замка: 0 - подчиниться защитному замку; 1 - игнорировать защитный замок

SF - системный флажок. Значение, записанное в этот разряд, записывается в соответствующий разряд регистра состояния

EAI - резервирован. Должен быть равен нулю

EKI -разрешение прерывания по заполнению выходного буфера. Нулевое значение означает «не генерировать прерывание», единичное «генерировать прерывание» при заполнении выходного буфера.

3). *Специальное чтение (A5h)* - 8042 помещает в выходной буфер истинное значение порта 2 за исключением разрядов 4 и 5, которым дается новое определение. Не генерируется состояние «Буфер полон». Если разряд 5 = 0, используется 9-разрядная клавиатура (XT); 5 = 1, используется 11-разрядная клавиатура; 4 = 0, запрещается прерывание по заполнению выходного буфера; 4 = 1, разрешается прерывание по заполнению выходного буфера.

4). *Инициализация(AAh)* - 8042 инициализирует порты 1 и 2, запрещает клавиатуру и очищает указатели буфера. Затем он помещает 55h в выходной буфер.

5). *Контроль интерфейса (ABh)*- направляет 8042 на тестирование линий данных и синхронизации интерфейса клавиатуры. Выходной буфер (порт ввода (60h) получает результаты тестирования:

00h - ошибок не обнаружено;

01h - линия синхронизации клавиатуры «остается нулевой»;

02h - линия синхронизации клавиатуры «остается единичной»;

03h - линия данных клавиатуры «остается нулевой»;

04h - линия данных клавиатуры «остается единичной».

Примечание: «остается нулевой» не проверяется для 9-разрядных клавиатур.

6). *Дамп диагностирования (AC h)*. Резервируется для диагностирования.

7). *Запретить клавиатуру (ADh)*- устанавливает разряд 4 командного байта 8042, который запрещает интерфейс клавиатуры. До разрешения клавиатуры данные не передаются и не принимаются.

8). *Разрешить клавиатуру (AEh)*. Сбрасывает разряд 4 командного байта 8042, который разрешает интерфейс клавиатуры.

9). *Прочитать входной порт (C0 h)*. Настраивает 8042 на передачу содержимого входного порта и помещение его в выходной буфер (порт ввода 60h). Пользуйтесь этой командой только при пустом выходном буфере.

10). *Прочитать выходной порт (D0h)*. Настраивает 8042 на передачу текущего байта выходного порта в выходной буфер (порт ввода 60h). Пользуйтесь этой командой только при пустом выходном буфере.

11). *Записать в выходной порт (D1h).* Помещает следующий байт, записанный по адресу 60h, в выходной порт. Разряд 0 выходного порта подключен к линии системного сброса. Этот разряд не должен переключаться из «единицы» в «нуль».

12). *Прочитать входы TEST0 и TEST1.* Настраивает 8042 на передачу текущего состояния входов TEST0 и TEST1 в выходной буфер (порт 60h) разряды 0 и 1 соответственно.

13). *Пульсация выходного порта (F0h - FFh).* Может быть вызвана пульсация разрядов 0 - 3 выходного порта (перевод в нулевое состояние примерно на 2 - 6 мкс.) Разряды 0 - 3 в этой команде указывают, какой разряд выходного порта должен пульсировать. Пульсация нулевого разряда приводит к сбросу системы.

7.4. Подготовка к лабораторной работе

⌚ Ознакомиться с описанием лабораторной работы «Программирование клавиатуры»

⌚ отладчиком AFD

⌚ и примерами программ (см. Приложение к лабораторной работе).

7.5. Порядок выполнения лабораторной работы

7.5.1.1 Написать на ассемблере программу посылки команды клавиатуре и определения кода подтверждения клавиатуры.

7.5.1.2 Написать программу зажигания индикаторов NumLock, CapsLock, ScrollLock.

7.5.1.3 Написать программу установки повтора кода клавиши с частотой 0,3 сек.

7.5.2.1 Написать программу чтения данных выходного порта.

7.5.2.2 Написать программу записи значения выходного порта.

7.5.2.3 Написать программу чтения данных входного порта.

7.5.2.4 Написать программу чтения портов состояния входных линий T0 и T1.

7.5.2.5 Написать программу сброса выходного порта.

Запустить и отладить программы в среде отладчика AFD.

Результат работы показать преподавателю.

Оформить отчет по результатам лабораторной работы.

7.6. СПИСОК ЛИТЕРАТУРЫ

1. AT BOOK (Текстовый файл формата DOS объемом 1.2 Мбайт).

2. «Аппаратура персональных компьютеров и ее программирование. IBM PC/XT/AT и PS/2». М: Радио и связь, 1995 г.

3. Финогенов К.Г. Самоучитель по системным функциям MS DOS. М: Радио и связь, 1995 г.

7.7. ПРИЛОЖЕНИЯ.

Примечание: в программах преднамеренно допущены неточности, которые необходимо устранить во время отладки программ.

7.7.1. Примеры программ:

Работа с клавиатурой

Программа 1. Включить все светодиоды. В результате программы включаются индикаторы NumLock, CapsLock, ScrollLock.

```
MOV    DX,0064          - ждем, пока входной буфер не пуст
A1: IN     AL,DX
```

	TEST	AL,00	
	JNZ	A1	
	MOV	AL,ED	- команда установки и сброса индикаторов
состояния	MOV	DX,60	
	OUT	DX,AL	
A2:	IN	AL,DX	- команда подтверждения приема
	CMP	AL,FA	
	JNZ	A2	
	MOV	AL,07	- установка индикаторов
	OUT	DX,AL	
A3:	IN	AL,DX	- команда подтверждения приема
	CMP	AL,FA	
	JNZ	A3	
	INT3		

Программа 2. Повторение клавиши с частотой 0,3 сек и начальной задержкой 0,5 с.

	MOV	DX,60	- команда задания частоты повтора кода клавиши
	MOV	AL,F3	
	OUT	DX,AL	
A1:	IN	AL,DX	- команда подтверждения приема
	CMP	AL,FA	
	JNZ	A2	
	MOV	AL,3B	- установка повторения клавиши
	OUT	DX,AL	
A2 :	IN	AL,DX	- команда подтверждения приема
	CMP	AL,FA	
	JNZ	A3	
	INT 3		

Работа с контроллером клавиатуры.

Программа 3. Чтение входного порта.

	MOV	DX,64
A1:	IN	AL,DX
	TEST	AL,02
	JNZ	A1
A2:	IN	AL,DX
	TEST	AL,01
	JNZ	A2
	MOV	AL,C0
A3:	IN	AL,DX
	TEST	AL,01
	JZ	A3
A4:	OUT	DX,AL
	TEST	AL,02
	LOOPNZ	A4
	MOV	DX,60
	IN	AL,DX
	INT3	

Программа 3. Запись в выходной порт.

```

      MOV     DX,64
A1:   IN      AL,DX
      TEST    AL,02
      JNZ     A1
A2:   OUT     DX,AL
      TEST    AL,02
      LOOPNZ  A2
      MOV     AL,00
      MOV     DX,60
A3:   OUT     DX,AL
      TEST    AL,02
      LOOPNZ  A3
      MOV     DX,64
A4:   IN      AL,DX
      TEST    AL,01
      JNZ     A4
      MOV     AL,D0
      OUT     DX,AL
      INT 3

```

Программа 4. Чтение состояния входных линий.

```

      MOV     DX,64
A1:   IN      AL,DX           - входной буфер полон?
      TEST    AL,02
      JNZ     A1
      MOV     AL,E0          - команда чтения состояния входных линий
A2:   OUT     DX,AL
A3:   IN      AL,DX
      TEST    AL,01          -ждать, пока заполнится выходной буфер
      JZ      A3
      MOV     DX,60          - чтение состояния входных линий
      IN      AL,DX
      INT 3

```

Программа 5. Пульсация выходного порта.

```

      MOV     DX,64
A1:   IN      AL,DX           - входной буфер полон?
      TEST    AL,02
      JNZ     A1
      MOV     AL,FE          - команда пульсации 0-вого разряда
      OUT     DX,AL          выходного порта
A2:   TEST    AL,02          - ждать, пока входной буфер освободится
      JNZ     A2
      INT 3

```

Лабораторная работа 7
(описание)
8 . «ПРОГРАММИРОВАНИЕ RTC CMOS»

8.1 . ЦЕЛЬ РАБОТЫ.

Целью лабораторной работы «Программирование RTC CMOS» является ознакомление с принципами работы контроллера типа MC146818 (Motorola) в качестве часов реального времени RTC (real-time clock) и памяти конфигурации системы, а также с приемами его программирования.

8.2 . ТЕХНИЧЕСКИЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ В РАБОТЕ.

В лабораторной работе используется :
ПЭВМ IBM PC AT и старше.

8.3 . НАЗНАЧЕНИЕ МИКРОСХЕМЫ MC146818(RTC/CMOS).

В архитектуре PC AT используется контроллер типа MC146818 (Motorola) в качестве часов реального времени RTC (real-time clock) и памяти конфигурации системы. Контроллер содержит 64 байт памяти (CMOS): первые 14 байт памяти используются непосредственно для внутренней схемы RTC, остальные 50 байт используются для размещения информации о конфигурации системы.

8.4 . ФУНКЦИОНИРОВАНИЕ МИКРОСХЕМЫ MC146818(RTC/CMOS).

MC146818 продолжает счет времени после отключения системного питания, при этом должен производиться переход на альтернативный источник питания микросхемы (обычно батареи). Во время этого перехода с системного питания на батарейное проектировщик системы RTC с батарейной поддержкой должен обеспечить защиту целостности данных, минимизацию энергопотребления и надежность аппаратных средств.

Программа процессора получает данные времени и календаря путем чтения соответствующих байтов памяти. Путем записи этих байтов в соответствующие регистры RAM программа может инициализировать часы, календарь и будильник. Содержимое 10 байтов часов, календаря и будильника может быть двоичным или двоично-десятичным (BCD). Адреса байтов часов, календаря, будильника, служебных регистров и т.д. подробно описаны в приложении 8.9.1.

Перед инициализацией внутренних регистров разряд SET регистра В должен быть установлен в 1, чтобы предотвратить изменения данных часов/календаря.

Программа инициализирует 10 ячеек в выбранном формате (двоичном или BCD), затем указывает этот формат в разряде кода режима данных (DM) регистра В.

Во всех 10 байтах часов, календаря и будильника должны быть использованы одинаковые данные - либо двоичные либо BCD. Для разрешения корректировок должен быть сброшен разряд SET. После инициализации RTC производит все корректировки данных в выбранном режиме. Для изменения режима необходимо реинициализировать указанные выше 10 байтов. Цикл корректировки проводится MC146818 каждую секунду при условии, что использована одна из определенных частот, делитель DV0-DV2 не сброшен, а разряд SET в регистре В сброшен. Если разряд SET установлен в 1, то программа может инициализировать байты часов и календаря; для этого прекращается текущий цикл корректировки и запрещаются новые циклы.

При частоте 4,194304 МГц или 1,048576 МГц цикл корректировки занимает 248 мкс, а при временной оси 32,768 кГц - 1984 мкс. Во время цикла корректировки программа

процессора не имеет доступа к байтам времени, календаря и будильника. MC146818 не позволяет программе читать промежуточные (переходные) данные. Эта защита достигается за счет отключения части RAM, относящейся к часам, календарю и будильнику, от шины микропроцессора на все время цикла корректировки. Если процессор будет читать эти байты RAM до завершения корректировки, выход будет неопределенным.

В это время устанавливается разряд состояния цикла корректировки UIP. При случайном обращении к данным о времени и дате доступ программы к этим данным оказывается невозможным ни в одной из 4032 попыток.

Разряд 24/12 в регистре В определяет, как ведется отсчет времени: 1-12 или 0-23. Этот разряд не может быть изменен без инициализации байта часов. При выборе 12-часового формата старший разряд байта часов соответствует времени после полудня при установке в 1.

Три байта будильника могут быть использованы двояко. Во-первых, когда программа устанавливает время будильника в соответствующих байтах часов минут и секунд, каждый день в указанное время вырабатывается прерывание будильника, если установлен разряд разрешения прерывания. Кроме того, программа может установить «безразличное» состояние в одном или всех байтах будильника. Кодом «безразличного» состояния служит любой шестнадцатеричный байт от C0 до FF, т.е. «безразличное» состояние вызывается, когда установлены в 1 два старших разряда каждого байта. Если «безразличный» код установлен в байте часов будильника, прерывание будильника вырабатывается каждый час. Если «безразличный» код установлен в байтах часов и минут будильника, прерывание будильника вырабатывается каждую минуту. Наконец, если «безразличный» код установлен во всех трех байтах будильника, прерывание вырабатывается каждую секунду.

RTC и RAM обеспечивают три отдельных автономных источника прерываний процессора. Прерывание будильника может быть запрограммировано на периодичность от одного раза в секунду до одного раза в день. Периодические прерывания могут быть выбраны с периодом от 0,5 с до 30,517 мкс. Для извещения программы о завершении цикла корректировки может быть использовано прерывание конца корректировки. Нужный режим прерывания выбирает программа процессора. Три разряда в регистре В разрешают три прерывания. При записи единицы в разряд разрешения прерывания прерывание будет выработано при наступлении соответствующего события. Нуль в разряде разрешения прерывания запрещает выводу IRQ реагировать на данное прерывание.

В MC146818 предусмотрено 22 двоичных каскада делителя по временной оси, как показано на блок-схеме (приложение рис. 1). Выход делителей представляет собой сигнал частотой 1 Гц, подающийся на логическое устройство, обеспечивающее цикл корректировки. Делители управляются тремя разрядами (DV2, DV1 и DV0) регистра А.

Делители могут иметь три базовых частоты (4,194304 МГц, 1,048576 МГц или 32,768 кГц). Цепочка делителя может удерживаться сброшенной, что дает возможность точной установки времени. Когда делитель переходит от сброса к рабочей временной оси, через 0,5 с осуществляется первый цикл корректировки. Разряды управления делителем используются также для облегчения тестирования MC146818.

У MC146818 присутствует выход прямоугольных сигналов, но он не используется в архитектуре PC AT.

Пятнадцать из 22 выходов делителя доступны для селектора 1-15, как показано на блок-схеме. Первая цель выбора выхода делителя состоит в генерации прямоугольного выходного сигнала на выводе SQW. Частота прямоугольного сигнала устанавливается

разрядами RS0-RS3 в регистре A.

Селектор 1-15 используется не только для выбора частоты SQW, но и для периодических прерываний.

После того как частота выбрана, программа управляет включением и выключением выходного сигнала на выводе SQW с помощью разряда разрешения прямоугольного сигнала (SQWE) в регистре B. Во время изменения разрядов выбора прямоугольного сигнала или разрешения прямоугольного сигнала SQWE может возникать сигнал асимметричной формы. Вывод SQW прямоугольного сигнала может использоваться по-разному. Например, он может служить эталоном частоты для внешних применений, синтезатором частоты или использоваться для генерирования одного или более звуковых тонов под управлением программы. В приложении 8.9.1 приведена блок-схема контроллера RTC и назначение выводов микросхемы.

8.5. ПРОГРАММИРОВАНИЕ МИКРОСХЕМЫ MC146818(RTC/CMOS).

Для чтения и записи во внутреннюю память RTC используются команды микропроцессора IN и OUT. Порт 70h используется совместно регистром маски NMI и регистром адресов памяти конфигурации памяти. Чтобы оставить разрешенным прерывание NMI, нужно, чтобы разряд 7 адресов RTC был установлен в единицу при записи в порт 70h(т.е. 1XXXXXXX, где XXXXXXXX - реальный адрес).

Чтобы записать значение в память RTC необходимо: использовать OUT 70h, AL -для указания адреса памяти конфигурации, которая будет изменена (70h - номер порта, AL - адрес памяти), с учетом вышеуказанного условия для NMI;

использовать OUT 71h, AL для записи данных в память RTC по адресу, указанному в предыдущей команде записи в порт 70h.

Команды вывода слов (например, OUT 70h, AX) для одновременной загрузки ячейки памяти и данных не допускаются.

Чтобы прочитать содержимое памяти конфигурации необходимо: использовать OUT 70h, AL для указания адреса памяти конфигурации, который будет читаться (70h - номер порта, AL - адрес памяти);

использовать IN AL, 71h для чтения данных, хранящихся по этому адресу.

Все 64 регистра памяти RTC доступны для чтения и записи, кроме следующих:

- регистров состояния C и D, доступных только для чтения;
- разряда 7 регистра состояния A, доступного только для чтения;
- старшего разряда байта секунд, доступного только для чтения;
- не используемого байта 3Fh.

В MC146818 имеются четыре служебных регистра(A,B,C,D), доступных программе процессора. Эти четыре регистра полностью доступны и во время цикла корректировки. Назначение битов служебных регистров смотри в приложении 8.9.1.

8.6 . УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ.

Выполнение лабораторной работы состоит из трех этапов:

- ⌚ подготовка к лабораторной работе;
- ⌚ проведение работ по непосредственному изучению RTC/CMOS принципов ее работы, программной настройки, написание и отладка программ, иллюстрирующих работу RTC/CMOS;
- ⌚ оформление отчета и сдача(защита) лаб. работы преподавателю.

8.6.1. Подготовка к лабораторной работе заключается в изучении описания лабораторной работы «Подсистема часов реального времени RTC/CMOS», описания работы микросхемы MC146818(RTC/CMOS), отладчика AFD, ознакомления с примерами программ(см. приложение). Подготовка к работе проводится студентом самостоятельно во внеурочное время.

8.6.2. Проведение работы по изучению принципов организации и работы RTC/CMOS разбивается на два этапа.

На первом этапе определяются значения управляющих слов служебных регистров RTC/CMOS и их адреса, формат представления времени в RTC, изучается текст резидентной сервисной программы BEEP_TSR.EXE (приведенный в приложении). На втором этапе производится создание и отладка программ, иллюстрирующих возможности программирования RTC/CMOS.

8.6.3. Оформление отчета по лабораторной работе осуществляется во внеаудиторное время. Сдача (защита) лабораторной работы производится во время выполнения следующей работы или во внеурочное время по согласованию с преподавателем.

8.6.4. Отчет по лабораторной работе должен содержать следующий материал:

- постановку задачи выполняемой лабораторной работы;
- краткое описание работы RTC/CMOS, возможностей подсистемы реального времени и значения управляющих слов служебных регистров;
- результаты изучения программирования различных функций RTC/CMOS;
- тексты программ на ассемблере;
- выводы о проделанной работе.

Отчет должен быть оформлен грамотно и аккуратно !

8.7 . ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ.

8.7.1. По описанию лабораторной работы определить адреса служебных регистров RTC/CMOS, адреса байтов времени, будильника, даты и т.п.

8.7.2. Определите управляющие слова для программирования различных функций RTC/CMOS: прерывание будильника, периодическое прерывание, генерация прямоугольных импульсов, перевод и установка даты/времени.

8.7.3. Запустите имеющуюся на диске сервисную резидентную программу BEEP_TSR.EXE для получения звуковых сигналов при возникновении прерывания от RTC/CMOS.

7.8.4. Используя программу 1, установите системную дату равной дате своего рождения(или дате указанной преподавателем). Результат покажите преподавателю.

8.7.5. Изучив текст программы 2, установите безразличный код в байтах будильника для получения прерывания раз в секунду. Результат покажите преподавателю.

8.7.6. Изучив текст программы 3, запрограммируйте RTC на генерацию периодического прерывания с частотой, указанной преподавателем. Результат покажите преподавателю.

8.8 . РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.

- 1) Лекции по курсу «Адаптеры и контроллеры ЭВМ».
- 2) Электронный учебник «PC Book»
- 3). «Аппаратура персональных компьютеров и ее программирование. IBM PC/XT/AT и PS/2». М: Радио и связь, 1995 г.

8.9 . ПРИЛОЖЕНИЯ.

8.9.1. Описание микросхемы MC146818(RTC/CMOS).

В архитектуре PC AT используется контроллер типа MC146818 (Motorola) в качестве часов реального времени RTC (real-time clock) и памяти конфигурации системы. Контроллер содержит 64 байт памяти (CMOS): первые 14 байт памяти используются непосредственно для внутренней схемы RTC, остальные 50 байт используются для размещения информации о конфигурации системы. На рис. 1 приведена блок-схема контроллера RTC, а в Таблице 1 - назначение выводов RTC.

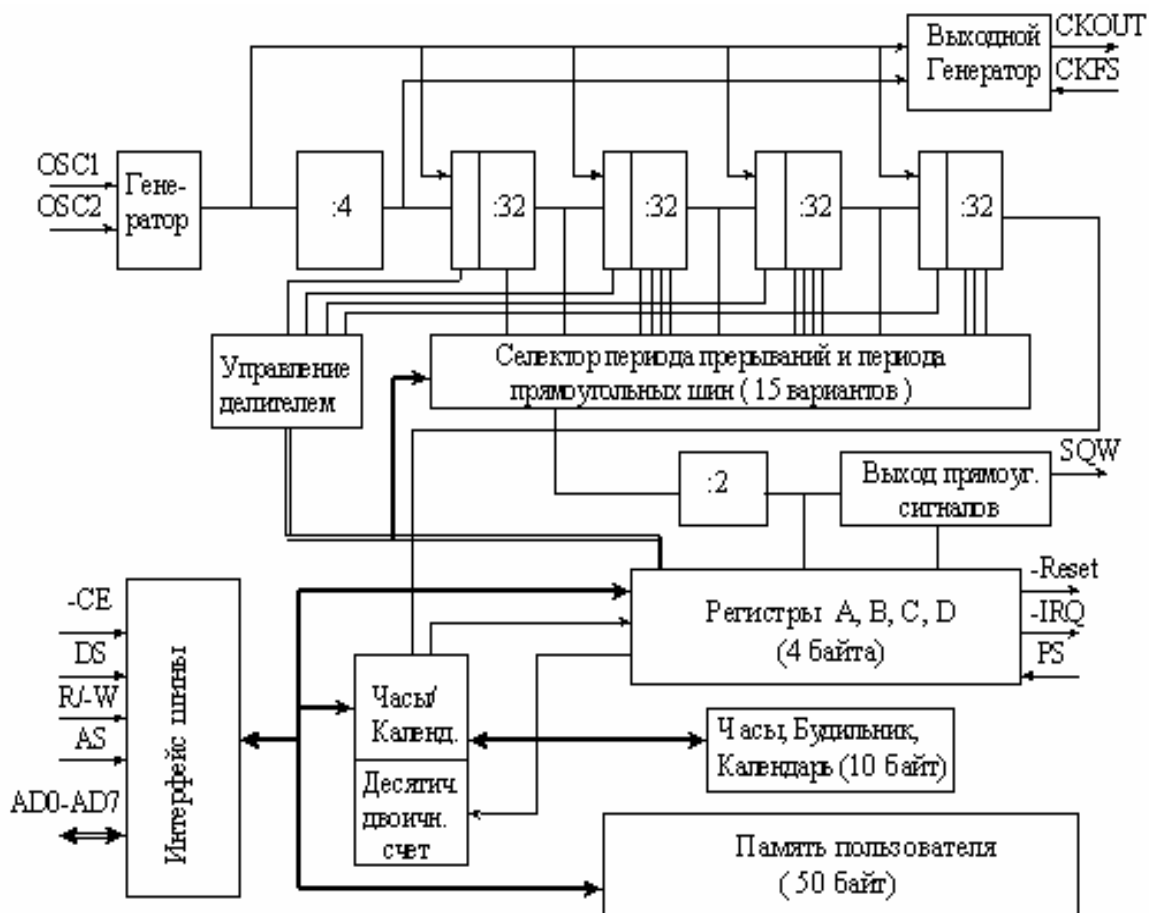


Рис.1. Блок-схема контроллера RTC.

Таблица 1. Назначение выводов RTC.

Обозначение вывода	Тип вывода	Номер вывода	Функциональное назначение
OSC1, OSC2	Вход	2, 1	Задающий генератор. Задающим генератором для временных функций может служить внешний сигнал или кварцевый генератор. К OSC1 могут быть подключены внешние источники прямоугольных сигналов с частотой 4,194304 МГц, 1,048576 МГц или 32,768 кГц. Частоту внутреннего генератора выбирает с помощью регистра A. Встроенный генератор спроектирован для параллельно-резонансного кварцевого кристалла AT-среза на частотах 4,194304 МГц или 1,048576 МГц
CKOUT	Выход	21	Выходной CLOCK. Это выход с частотой внутреннего генератора, деленной на 1 или на 4. Основное применение CKOUT – входной CLK для микропроцессора в целях снижения стоимости второго кристалла. Частота CKOUT зависит от частоты внутреннего генератора и состояния вывода CKFS
CKFS	Вход	20	Выбор частоты CKOUT. Когда CKFS подключен к VDD, частота сигнала на CKOUT идентична частоте сигнала на входе OSC1. Когда CKFS подключен к VSS, частота сигнала на CKOUT в четыре раза ниже частоты временной оси. В PC AT этот вывод подключен к VDD
SQW	Выход	23	Прямоугольные сигналы. На SQW может быть выведен сигнал от одного из 15 выходов 22-каскадного внутреннего делителя. Частота сигнала на выходе SQW может изменяться путем программирования регистра A. Для включения и выключения сигнала SQW может использоваться разряд SQWE в регистре B. В PC AT этот вывод не используется
AD0-AD7	Вход-выход	4-11	Мультиплексная двунаправленная шина адреса/данных. В процессорах с мультиплексными шинами адрес выдается в первой части цикла шины, а данные – во второй части цикла шины. Мультиплексирование "данные вслед за адресом" не увеличивает время доступа MC146818, поскольку реверсирование шины от адреса к данным происходит во время внутреннего обращения к RAM. Адрес должен быть действительным сразу перед понижением AS/ALE, когда MC146818 защелкивает адрес из AD0-AD6. Действительные данные записи должны быть представлены и удержаны в течение заднего (положительного) фронта DS или WR. В цикле чтения MC146818 выводит 8 разрядов данных по заднему фронту DS или RD. Шина AD0-AD7 переходит в третье состояние, когда DS или RD становятся не активными

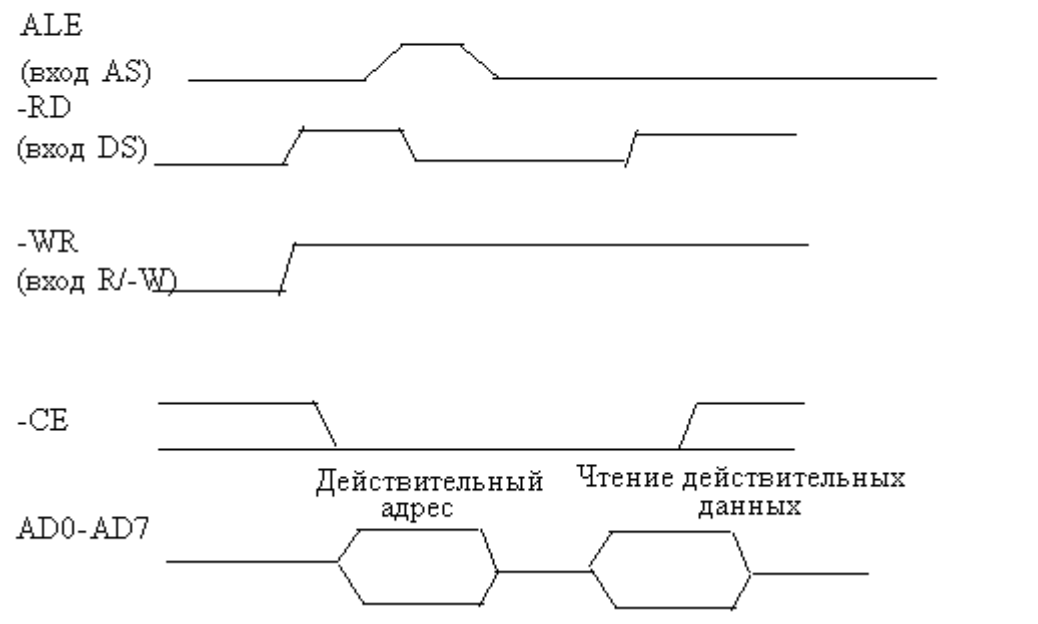
AS	Вход	14	Мультиплексный адресный строб. Положительный мультиплексный адресный строб-импульс служит для демultipлексирования шины. По заднему фронту AS или ALE адрес защелкивается в MC146818. Автоматическая схема MOTEL в MC146818 защелкивает также состояние вывода DS по заднему фронту AS или ALE
DS	Вход	17	Строб данных или чтение. DS может интерпретироваться схемой MOTEL двумя способами. При генерации от процессора типа Motorola DS представляет собой положительный импульс во время последней части цикла шины и может называться DS (строб данных), E (разрешение) и $\phi 2$ (clock $\phi 2$). Во время циклов чтения DS указывает время, когда RTC должен возбудить двунаправленную шину. В циклах записи по заднему фронту DS RTC и RAM защелкивают записанные данные. Вторая интерпретация DS схемой MOTEL - это RD, MEMR или IOR, генерируемые от процессора конкурирующего типа. В этом случае DS идентифицирует период времени, когда RTC и RAM возбуждают шину с данными чтения. Эта интерпретация DS соответствует сигналу разрешения выхода в обычной памяти. В архитектуре PC AT реализована вторая интерпретация. Схема MOTEL в MC146818 защелкивает состояние вывода DS по заднему фронту AS/ALE, уровень DS должен оставаться высоким при высоком уровне AS/ALE
IRQ	Выход	19	Запрос прерывания. Активный уровень - низкий, при котором он может быть использован как вход прерывания. Выход IRQ остается низким, пока установлен разряд состояния, вызывающий прерывание, и соответствующий разряд разрешения прерывания. Для сброса IRQ программа процессора обычно читает регистр C. Вывод RESET также сбрасывает "висящий" запрос на прерывание. Когда ситуация прерывания отсутствует, IRQ находится в высокоимпедансном состоянии

R/W	Вход	15	Чтение/запись. R/W – это отрицательный импульс записи MEMW или IOW от процессора. В этом режиме схема MOTEL рассматривает R/W как импульс записи (W) во многих RAM
CE	Вход	13	Разрешение кристалла. Активный уровень – низкий. Он должен быть активным во время действия DS и AS или RD и WR (в случае MOTEL). Циклы шины, в которых CE не активен, не вызывают никаких действий в MC146818. При высоком уровне CE выход мультиплексной шины находится в третьем состоянии. При высоком уровне CE все входы адресов, данных, DS и R/W от процессора отключаются в MC146818. Это позволяет изолировать MC146818 от процессора с выключенным питанием. Когда CE удерживается на высоком уровне, обесточенное устройство не может получить питание через входные выводы от источника питания RTC. Потребление энергии батарей может быть снижено с помощью "повышающего" резистора или активного фиксатора CE при выключенном системном источнике питания
RESET	Вход	18	Сброс. RESET не влияет на функции часов, календаря или RAM. При включении питания RESET должен удерживаться на низком уровне в течение определенного времени (минимум 5 мкс), чтобы была возможной стабилизация источника питания. При низком уровне RESET происходит следующее: а) разряд разрешения периодического прерывания (PIE) сбрасывается в нуль; б) разряд разрешения сигнального прерывания (AIE) сбрасывается в нуль; в) разряд разрешения прерывания по окончании корректировки (UIE) сбрасывается в нуль; г) разряд флага разрешения прерывания по окончании корректировки (UF) сбрасывается в нуль; д) разряд флага состояния запроса прерывания (IRQF) сбрасывается в нуль; е) разряд флага периодического прерывания (PF) сбрасывается в нуль; ж) разряд флага сигнального прерывания (AF) сбрасывается в нуль;

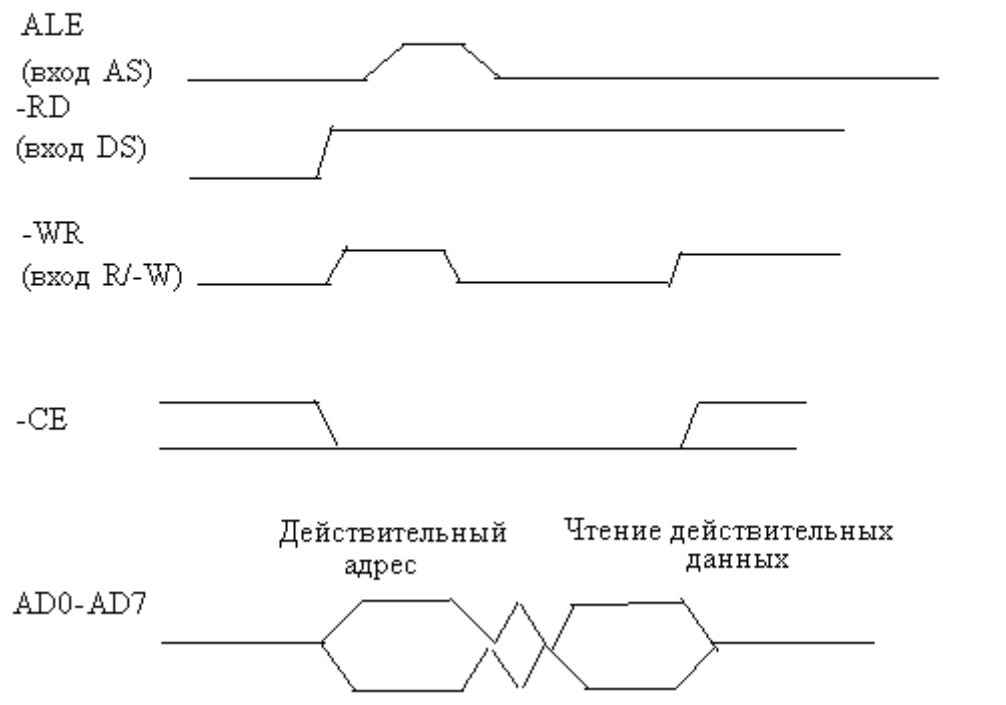
			з) вывод IRQ находится в высокоимпедансном состоянии; и) разряд разрешения квадратного сигнала выхода (SQWE) сбрасывается в нуль
PS	Вход	22	Состояние питания. PS используется для контроля действительности содержимого RAM и времени (разряд VRT) в регистре D. При низком уровне вывода PS разряд VRT сбрасывается в нуль. При использовании функции VRT во время включения питания вывод PS должен удерживаться ниже на низком уровне в течение определенного времени (минимум 5 мкс), что соответствует номинальному значению напряжения батареи. Если батарея отсутствует, то при включении питания уровень VRT остается низким, указывая, что содержимое RAM, регистров времени и календаря не гарантировано. После включения питания уровень PS должен повыситься, что позволяет установить разряд VRT при чтении регистра D. В PC AT этот вывод соединен с выводом RESET
VDD, VSS	Входы	24, 12	На эти два вывода подается питание (постоянный ток), причем напряжение VDD выше, чем VSS. Значения напряжения VDD: (-0,3 ... +8,0) В

Временные диаграммы чтения/записи мультиплексированной шины.

Временная диаграмма чтения мультиплексированной шины.



Временная диаграмма записи мультиплексированной шины.



Карта адресов памяти МС 146818

Адрес регистра	Функциональное назначение
00h	Секунды
01h	Секунды будильника
02h	Минуты
03h	Минуты будильника
04h	Часы
05h	Часы будильника
06h	День недели
07h	Число месяца
08h	Месяц
09h	Год
0Ah	Регистр состояния A
0Bh	Регистр состояния B
0Ch	Регистр состояния C
0Dh	Регистр состояния D
0Eh	Байт диагностирования
0Fh	Байт кода сброса
10h	Тип НГМД (A или B)
11h	Резерв
12h	Тип НМД (C или D)
13h	Резерв

14h	Установленное оборудование
15h, 16h	Объем базовой памяти
17h, 18h	Объем расширенной памяти (более 1 М байт)
19h	Тип НМД 1
1Ah	Тип НМД 2
1Bh-2Ch	Резерв
2Dh	Дополнительный флажок
2Eh-2Fh	Контрольная сумма содержимого адресов 10h-2Dh
30h-31h	Расширенная память
32h	Текущий век в коде BCD
33h	Системная информация
34h-3Eh	Резерв
3Fh	Не используется

Режимы службы времени RTC

Адрес байта	функция	Десятичный интервал	Интервал	
			Режим двоичных данных	Режим десятичных данных
0	Секунды	0-59	00-3B	00-59
1	Секунды будильника	0-59	00-3B	00-59
2	Минуты	0-59	00-3B	00-59
3	Минуты будильника	0-59	00-3B	00-59
4	Часы (12-часовой режим) Часы (24-часовой режим)	1-12 0-23	01-0C (AM) и 81-8C (PM) 00-17	01-12 (AM) и 81-92 (PM) 00-23
5	Часы будильника (12-часовой режим) Часы будильника (24-часовой режим)	1-12 0-23	01-0C (AM) и 81-8C (PM) 00-23	01-12 (AM) и 81-92 (PM) 00-23
6	День недели (воскресенье = 1)	1-7	01-07	01-07
7	День месяца	1-31	01-1F	01-31
8	Месяц	1-12	01-0C	01-12
9	Год	0-99	00-63	00-99

Служебные регистры

В MC146818 имеются четыре служебных регистра, доступных программе процессора. Эти четыре регистра полностью доступны и во время цикла корректировки.

Регистр А (0A).

Чтение/запись, кроме UIP.

7							0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

UIP - разряд текущего цикла корректировки - флажок состояния, который может контролироваться программой. Когда UIP установлен в 1, корректировка происходит или должна начаться. Когда UIP установлен в нуль, цикла корректировки нет и он не начнется раньше, чем по меньшей мере через 244 мкс. Данные часов, календаря и будильника в RAM полностью доступны программе при UIP=0, в это время они не изменяются. Разряд UIP доступен только для чтения и не управляется выводом RESET. При установке SET в регистре В в 1 отсутствует цикл корректировки, и UIP сбрасывается.

DV2, DV1, DV0 - три разряда, позволяющие программе выбирать различные режимы 22-каскадного делителя. Разряды выбора делителя идентифицируют используемую частоту. Может быть использована частота 4,194304 МГц, 1,048576 МГц и 32,768 кГц. Разряды выбора делителя используются также для сброса цепочки делителя. При инициализации часов календаря программа может запустить делитель в определенный момент, который задан в RAM. При снятии сброса делителя через одну секунду начинается первый цикл корректировки.

RS3, RS2, RS1, RS0 используются для выбора одного из 15 выходов 22-каскадного делителя или запрещения выхода делителя. Выбранный выход может использоваться для генерирования выходного прямоугольного сигнала (вывод SQW) и/или периодического прерывания. Программа может

- 1) разрешить прерывание с помощью PIE;
- 2) разрешить выход SQW с помощью SQWE;
- 3) разрешить одновременно прямоугольный сигнал и прерывание с одинаковой периодичностью;
- 4) не разрешить ни того, ни другого.

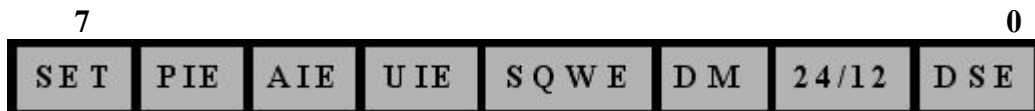
Указанные четыре разряда доступны для чтения и записи, на них не действует вывод RESET. Соответствие содержимого разрядов RS3, RS2, RS1, RS0 и частоты выходного прямоугольного сигнала и/или периодического прерывания приведено в таблице 4.

Таблица 4.

R S (hex)	F	E	D	C	B	A	9	8
F (Г ц)	2	4	8	16	32	64	128	256
R S (hex)	7	6	5	4	3	2	1	
F (Г ц)	512	1024	2048	4096	8129	16384	32768	

Регистр В (0В)

Чтение/Запись.



SET: когда этот разряд установлен в нуль, цикл корректировки функционирует обычным образом, инкрементируя счет каждую секунду. Когда SET установлен в 1, цикл корректировки запрещен. SET доступен для чтения и записи,

но он не изменяется выводом RESET или внутренними функциями MC146818.

PIE - разряд разрешения периодического прерывания, доступный для чтения и записи; он позволяет флажковому разряду периодического прерывания (PF) в регистре С понижать уровень вывода IRQ. Программа записывает 1 в PIE, чтобы получать периодические прерывания с частотой, заданной разрядами RS3, RS2, RS1 и RS0 в регистре А. Нулевой PIE блокирует возбуждение IRQ от периодического прерывания, но при этом остается установленным флажковый разряд PF. PIE не модифицируется никакими внутренними функциями MC146818, но сбрасывается в нуль выводом RESET.

AIE - разряд разрешения сигнального прерывания, доступен для чтения и записи; он разрешает флажковому разряду сигнализации (AF) устанавливать

IRQ. Сигнальное прерывание вырабатывается каждую секунду, когда три разряда времени совпадают с тремя разрядами сигнализации, включая «безразличный» код (11XXXXXX). Когда разряд AIE установлен в нуль, разряд AF не активизирует сигнал IRQ. Вывод RESET сбрасывает разряд AIE в нуль. Внутренние функции MC146818 не влияют на разряд AIE.

UIE - разряд прерывания конца корректировки, доступный для чтения и записи; он позволяет флажковому разряду конца корректировки (UF) в регистре С устанавливать IRQ. При понижении вывода RESET или при установке SET разряд UIE сбрасывается.

SQWE - разряд разрешения прямоугольного сигнала. Когда он установлен программой в 1, на вывод SQW выводится прямоугольный сигнал, частота которого выбрана разрядами RS3-RS0. Когда SQWE установлен в нуль, вывод SQW удерживается на низком уровне. SQWE сбрасывается выводом RESET. SQWE доступен для чтения и записи.

DM -разряд режима данных, показывающий, в каком формате должна выполняться корректировка данных часов и календаря - в двоичном или BCD. DM записывается программой процессора и может читаться программой, но он не модифицируется никакими внутренними функциями или RESET. Единица в DM соответствует двоичным данным, нуль - двоично-кодированным десятичным.

24/12 - разряд, устанавливающий формат байтов часов: 24-часовой режим(1) или 12-часовой режим (0). Этот разряд доступен для чтения и записи и изменяется только программно.

DSE - разряд разрешения летнего времени, доступный для чтения и записи при установке в 1 позволяет программе разрешить две специальные корректировки. В последнее воскресенье апреля время инкрементируется с 1.59.59 на 3.00.00. В последнее воскресенье октября время по достижении 1.59.59 изменяется на 1.00.00. Эти специальные корректировки не производятся, когда DSE установлен в нуль. DSE не изменяется никакими внутренними функциями или RESET.

Регистр С (0C).

Чтение/запись



IRQF -флажок запроса прерывания, устанавливается в 1, когда выполняется одно или несколько следующих условий:

PF = PIE = 1 ,

AF = AIE = 1 ,

UF = UIE = 1 ,

т. е. $IRQF = PF * PIE + AF * AIE + UF * UIE$

Когда IRQF установлен в 1, вывод IRQ находится на низком уровне. После чтения регистра С программой или после понижения вывода RESET все флажковые разряды сбрасываются.

PF - флажок периодического прерывания, доступный для чтения и записи. Он устанавливается в 1, когда обнаруживается определенный фронт на выбранном отводе цепочки делителя. Частота периодических прерываний задается разрядами RS3-RS0. PF устанавливается в 1 независимо от состояния PIE. При PF = 1 возбуждается сигнал IRQ, а если и PIE = 1, то устанавливается IRQF. Разряд PF сбрасывается по выводу RESET или при считывании программой регистра С.

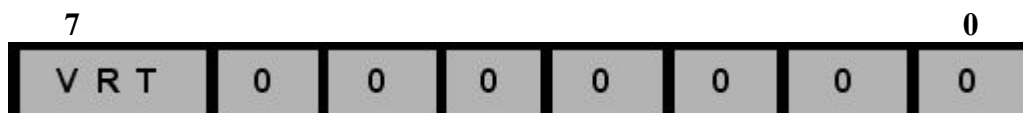
AF - флажок сигнального прерывания. Установка его в 1 показывает, что текущее время совпало со временем будильника. При этом понижается уровень вывода IRQ и IRQF устанавливается в 1, если AIE также установлен в 1. AF сбрасывается по RESET или при считывании программой регистра С.

UF - флажок конца корректировки, устанавливаемый после каждого цикла корректировки. Когда UIE установлен в 1, то при единице в UF устанавливается в 1 IRQF и возбуждается вывод IRQ. UF сбрасывается при чтении регистра С или по RESET.

Разряды 3-0 - неиспользуемые разряды регистра состояния С, читаются как нулевые, запись в них невозможна.

Регистр D (0D)

Только чтение.



VRT - разряд подтверждения корректности содержимого RAM и времени при условии правильного подключения вывода состояния питания (PS). При низком уровне PS в VRT появляется нуль. Программа процессора может установить VRT при инициализации часов и календаря, чтобы он указывал верные RAM и время. VRT доступен только для чтения и не изменяется по RESET. VRT может быть установлен только чтением регистра D.

8.9.2. Содержимое обработчика прерываний программы BEEP_TSR.EXE.

```

mov    al,0x0c;
out     0x70,al;
in      al,0x71      ; // сбрасываем прерывание от RTC
in      al,0x61      ; // далее программируем таймер

```

```

and    al,0xfc;
out    0x61,al;
mov    al,0xb7;
out    0x43,al;
mov    al,0;
out    0x42,al;
mov    al,20;
out    0x42,al;
in     al,0x61;
or     al,0x03;
out    0x61,al;
mov    cx,0xffff;
a: mov  dx,cx;           // два вложенных цикла для задержки
    mov  cx,0x0008;
b: loopnz b;
    mov  cx,dx;
    loopnz a;
    in   al,0x61;
    and  al,0xfc;
    out  0x61,al;
    mov  al,0x20;
    out  0xa0,al;       // сообщаем контроллерам, что прерывание закончено
    out  0x20,al;
    int  0x60           // вызываем старый обработчик прерываний от RTC
                        // заранее размещенный нами по свободному вектору 0x60

```

8.9.3. Примеры программ.

Программа 1.

Установка системной даты: 12 апреля 74 года.

```

mov    AL,8B
out    [70],AL
mov    AL,83 // устанавливаем SET
out    [71],AL
mov    AL,89
out    [70],AL
mov    AL,74 // год в BCD-коде
out    [71],AL
mov    AL,88
out    [70],AL
mov    AL,04 // месяц в BCD-коде
out    [71],AL
mov    AL,87
out    [70],AL
mov    AL,12 // день в BCD-коде
out    [71],AL
mov    AL,8B
out    [70],AL
mov    AL,83 // сбрасываем SET
out    [71],AL
int 3

```


Программа 2.

Установка будильника на генерацию прерываний каждый час через

15 минут 13 секунд.

```

mov  AL,8B
out  [70],AL
mov  AL,03  // устанавливаем SET
out  [71],AL
mov  AL,81
out  [70],AL
mov  AL,13  // секунды будильника BCD-коде
out  [71],AL
mov  AL,83
out  [70],AL
mov  AL,0F  // минуты будильника в BCD-коде
out  [71],AL
mov  AL,85
out  [70],AL
mov  AL,0F  // часы будильника в BCD-коде
out  [71],AL
mov  AL,8B
out  [70],AL
mov  AL,43  // сбрасываем SET и разрешаем прерывания будильника
out  [71],AL
int  3

```

Программа 3.

Установка RTC на генерацию периодических прерываний с частотой 16384 гц

```

mov  AL,8A
out  [70],AL
in   AL,[71]      // читаем содержимое регистра A
and  AL,F0
or   AL,06        // задаем частоту
out  [71],AL
mov  AL,8B
out  [70],AL
mov  AL,43        // разрешаем периодические прерывания
out  [71],AL
int  3

```

Программа 4.

Установка RTC на генерацию прямоугольного сигнала с частотой 16384 гц

```

mov  AL,8A
out  [70],AL
in   AL,[71]      // читаем содержимое регистра A
and  AL,F0
or   AL,02        // задаем частоту
out  [71],AL
mov  AL,8B

```

```
out    [70],AL
mov     AL,0B      // разрешаем генерацию прямоугольного сигнала
out     [71],AL
int 3
```

Программа 5.

Для запрета прерываний запустить из AFD следующую программу

```
mov     AL,AL
out     [70],AL
mov     AL,03
out     [71],AL
int 3
```

9. Приложение

9.1. Основные команды отладчика AFD (Advanced Fullscreen Debug)

L имя файла{,перем, адрес1}- загрузить файл в память. Адрес программы можно задавать(адрес1). По умолчанию -адрес CS:0100. Количество отображаемых байтов находится в BX, CX после окончания команды. Параметр имя_файла- это имя файла. Расширение (по умолчанию .exe).

Пример: L aaa.txt , , CS:0140

W имя файла, адрес, количество байт - записать данные в файл. Адресный сегмент по умолчанию -DS, количество_байт -определяет количество байт (4-х разрядное шестнадцатеричное число).

Пример: W aaa.txt ,0100,0136

{R} регистр=содержимое регистра - установить регистр. FL=содержимое регистра, флаговый регистр установится как 16 -разрядный регистр. Для раздельного доступа к флаговым битам нужно использовать OF, DF, IF, SF, ZF, AF, PF, CF.

D адрес - показать код. Указатель устанавливается в окне дизассемблера на этот адрес. Сегмент по умолчанию CS.

M n адрес

[регистр] - показать окно памяти (n=1 или n=2). Сегмент по умолчанию тот, который показан в соответствующем окне. FS можно использовать для фиксации памяти. Для косвенной адресации, содержимое любого регистра может использоваться, как адрес.

Пример: M 2 100

G {нач адрес}{,кон адрес}- выполнить программу с текущего адреса или со стартового адреса нач_адрес, кон_адрес - точка останова. Сегмент по умолчанию - текущий CS.

Пример: G 2aaa,2baa

A {адрес1}- переход в ассемблерный режим, в котором происходит редактирование программ. Инструкции вводятся по команде «Enter», клавиши управления курсором используются для перемещения между командами с области кодов вверх и вниз. адрес1 - адрес входа (по умолчанию -текущий адрес).

R адрес, строка - «взлом » памяти. Сегмент по умолчанию CS.

F адрес, число повторений, строка - заполнить память указанной строкой. Адресный сегмент по умолчанию DS. Число повторений определяет сколько раз строка будет записана в память.

S {адрес1}{,строка1} - поиск данных в памяти, начиная с адреса {адрес1}. Если адрес не указан, поиск начинается с CS:00. Когда данные обнаружены, экран отобразит эту область памяти. Команда S без параметров начинает повторный поиск. Строка1-строка символов, любой набор значений, 1234, BX, «ASCII», и т.д.

C адрес1, адрес2, длина - сравнение двух областей памяти. Если обнаружено несовпадение, то на экране M1 будет отражена область памяти, адресуемая первым параметром, а на экране M2 -область памяти, адресуемая вторым параметром. Сегмент по умолчанию DS.

CO адрес1, адрес2, количество байт - копирование данных из области памяти с адресом адрес1 в область памяти с адресом адрес2.

I адрес-отображение содержимого порта ввода/вывода. Адрес-любой 8-ми или 16-ти разрядный адрес или название регистра.

Пример: i 2fb i ax

O адрес, константа - вывод 8-разрядной константы в порт ввода/вывода, параметр адрес -см. команду I.

Пример: O 2fb ,00

T{B}- показать буфер трассировки. Если параметр B не указан,

использован отдельный метод отображения.

BW имя файла - записать установки точки останова в файл.

BL имя файла - считать установки точки останова из файла.

PH адрес, длина{, имя файла} - распечатать данный в шестнадцатичном и ASCII формате. Сегмент по умолчанию DS. Длина определяет количество байт. По умолчанию вывод на принтер.

PD адрес1, длина{, имя файла} - распечатать дизассемблированный код. Сегмент по умолчанию CS. Параметр длина - количество команд, которые будут распечатаны. Параметр имя_файла - имя файла, из которого необходимо взять команды для печати.

Пример: PD 100 , 10, aaa.asm

PT{смещение, длина{, имя файла}} - распечатать содержимое буфера трассировки. Смещение определяет смещение по 1 печатаемой команде. Число печатаемых инструкций определяется длиной или действительным числом записанных команд. По умолчанию - все команды.

286 ON

OFF - переключение режима ассемблера и дизассемблера для I80286.

Переключатель установлен на PC тип (PC/AT).

MO{DE} M{ONO}

C{OLOR}

A{LTERN} ON

OFF - установить режим экрана.

M - черно-белый адаптер.

C - цветной адаптер.

Если указано **A ON**, то выполнение программы будет сопровождаться отображением выводимых ею на экран данных (так называемый «теневого экран»). F6 - переключение между основным и теневым экраном. Команда без параметров возвращает текущие установки.

BE{EP} ON

OFF - включение и выключение звука.

XT - перейти в режим обучения. Все нажатия клавиш хранятся в буфере и могут быть записаны в файл или выполнены. Режим обучения завершается, когда буфер переполнен или по Ctrl+Break.

XX {имя файла} - выполнить записанные в режиме обучения нажатия клавиш. Если указано имя файла, то данные считываются из файла и выполняются.

XW имя файла - записать нажатия клавиш, хранящихся в буфере в файл.

XX имя файла - заполнить буфер клавиатуры данными из файла.

CTRL-HOME - выход из ассемблерного редактора в интерактивный режим для ввода команд.

CTRL-ESC - выход из бесконечного цикла (созданного, например командой JMP).

QUIT - выход из AFD и возврат в DOS.

F1 - выполнение команды

F2 - выполнение процедуры

F3 - восстановление команды

F4 - помощь

F5 - просмотр окна трассировки

F7 - перемещение вверх

F8 - перемещение вниз

F9 - перемещение влево

F10 - перемещение вправо

2. Система команд микропроцессора K1810BM86 (I8086)

2.1. Форматы команды

Микропроцессор K1810BM86 относится к классу однокристальных с фиксированной системой команд. При рассмотрении системы команд удобно обращаться к программной модели МП, содержащей его функциональные узлы (регистры), доступные программисту (рис. 2.1). Общие регистры разбиты на две группы: 1) группа HL, состоящая из регистров AX, BX, CX, DX, которые предназначены прежде всего для хранения данных и допускают раздельную адресацию их старших (H) и младших (L) половин; 2) группа PI, содержащая указательные регистры BP, SP и индексные регистры SI, DI, в которых обычно хранится адресная информация. Для общих и сегментных регистров указаны коды, используемые в форматах команд для их адресации. Регистр флагов F и указатель команд IP адресуются в командах неявно.

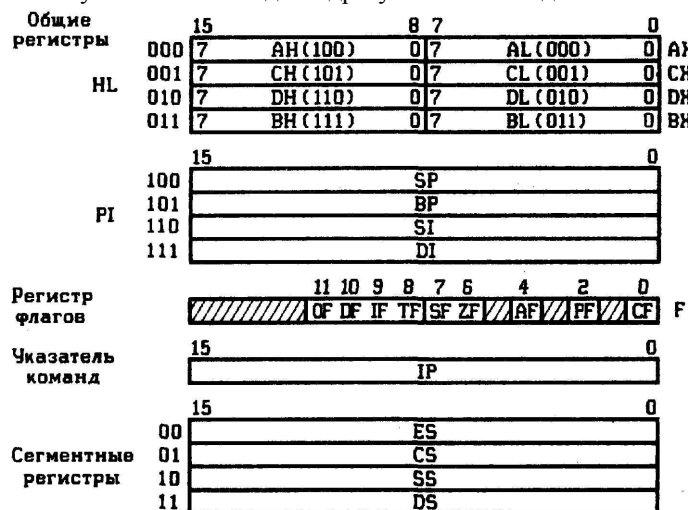


Рис. 2.1. Программно-доступные регистры МП ВМВБ

Команды МП ВМ86 могут адресовать один или два операнда, причем двухоперандные команды являются, как правило, симметричными, так как результат операции может быть направлен на место любого из операндов. Однако в таких командах один из операндов должен обязательно располагаться в регистре, поскольку имеются команды типа регистр — регистр, регистр — память и память — регистр, но команды типа память — память отсутствуют (за исключением команды пересылки цепочки байт или слов).

В общем виде формат двухоперандной команды приведен на рис. 2.2, а, где штриховыми линиями обозначены необязательные байты команды. Первый байт команды содержит код операции COP и два однобитовых поля: направления d и слова w. При d=1 осуществляется передача операнда или результата операции в регистр, который определяется полем рег второго байта команды; при d=0 — передача из указанного регистра. Поле w идентифицирует тип (разрядность) операндов: при w=1 команда оперирует словом, при w=0 — байтом.

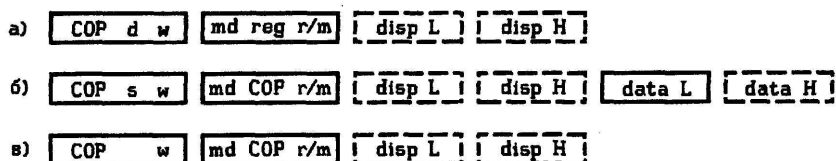


Рис. 2.2. Форматы двухоперандных команд

Второй байт, называемый постбайтом, определяет участвующие в операции регистры или регистр и ячейку памяти. Постбайт состоит из трех полей: md — режим, reg — регистр, r/m — регистр/память. Поле reg определяет операнд, который обязательно находится в регистре МП и условно считается вторым операндом. Поле r/m определяет операнд, который может находиться в регистре или памяти и условно считается первым. Способ кодирования внутренних регистров МП в полях reg и r/m представлен в табл. 2.1.

Отметим, что поле рег используется для указания регистра только в двухоперандных командах. Если в команде один операнд, то он идентифицируется полем r/m, а поле рег используется для расширения кода операции.

reg,		reg,			
r/m	w=0	w=1	r/m	w=0	w=1
000	AL	AX	100	AH	SP
001	CL	CX	101	CH	BP
010	DL	DX	110	DH	SI
011	BL	BX	111	BH	DI

Поле r/m	Эффективный адрес EA	Адресация
000	BX+SI+disp	Базово-индексная
001	BX+DI+disp	
010	BP+SI+diap	
011	BX+DI+disp	
100	SI+disp	Индексная
101	DI+diap	
110	BP+disp	Базовая
111	BX+disp	

Поле md показывает, как интерпретируется поле r/m для нахождения первого операнда: если md=ll, то операнд содержится в регистре, в остальных случаях — в памяти. Когда адресуется память, поле md определяет вариант использования смещения disp, находящегося в третьем и четвертом байтах команды:

- md = { 00, disp = 0 — смещение отсутствует;
 01, disp = disp L — команда содержит 8-битовое смещение, которое расширяется со знаком до 16 бит;
 10, disp = disp H, disp L — команда содержит 16-битовое смещение.

При md=ll реализуется косвенная адресация памяти и поле r/m определяет правила формирования эффективного адреса EA операнда в соответствии с табл. 2.2, где disp означает смещение, заданное в формате команды.

Приведенные в табл. 2.2 правила имеют одно исключение, позволяющее реализовать прямую (абсолютную) адресацию: если md=00 и r/m=110, то EA=disp H, disp L. Таким образом, имеется три варианта интерпретации поля md и восемь вариантов интерпретации поля r/m, что дает 24 варианта вычисления эффективного адреса EA. Суммарные сведения о постбайтовых режимах адресации ЦП ВМ86 приведены в табл. 2.3 и на рис. 2.3.

Поле		Поле md			
r/m	00	01	10	11	
				W = 1	W = 0
000	BX+SI	BX+SI+D8	BX+SI+D16	AL	AX
001	BX+DI	BX+DI+D8	BX+DI+D16	CL	CX
010	BP+SI	BP+SI+D8	BP+SI+D16	DL	DX
011	BP+DI	BP+DI+D8	BP+DI+D16	BL	BX
100	SI	SI+D8	SI+D16	AH	SP
101	DI	DI+D8	DI+D16	CH	BP
110	D16	BP+D8	BP+D16	DH	SI
111	BX	BX+D8	BX+D16	BH	DI

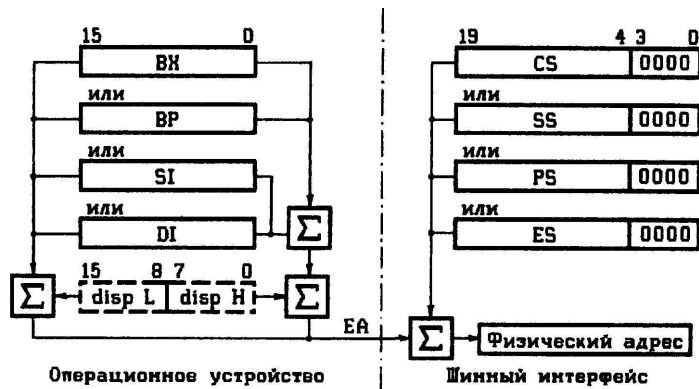


Рис. 2.3. Вычисление физического адреса памяти

Рассмотренная интерпретация полей *md* и *r/m* справедлива для всех форматов команд с постбайтовой адресацией.

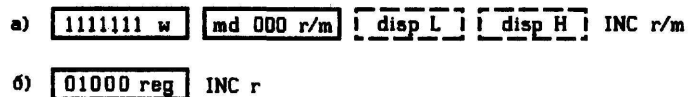
Подчеркнем смысловое различие двух случаев употребления термина “смещение”. Смещение *disp*, содержащееся в команде, интерпретируется как знаковое целое, которое участвует в вычислении эффективного адреса *EA*. С другой стороны, из-за сегментной организации памяти весь эффективный адрес *EA* является смещением (*offset*) относительно базового адреса сегмента и интерпретируется как беззнаковое целое при вычислении физического адреса. В необходимых случаях во избежание ошибок будем называть *offset* смещением в сегменте.

Наиболее общий формат двухоперандной команды с непосредственным операндом приведен на рис. 2.2,б. Необходимость адресации второго операнда отсутствует, и поле *reg* используется для расширения кода операции. Отсутствует также бит направления *d*, так как результат операции можно поместить только на место первого операнда. Место этого бита занимает бит *s*, который является признаком использования одного байта для задания непосредственного операнда при работе со словами. Поля *s* и *w* интерпретируются следующим образом:

$$SW = \begin{cases} x0, & \text{один байт данных data L;} \\ 01, & \text{два байта данных data H, data L;} \\ 11, & \text{один байт данных, который автоматически расширяется со знаком до 16 бит.} \end{cases}$$

Типичный формат однооперандной команды (рис. 2.2,б) содержит поля, назначение которых уже рассмотрено.

Следует отметить, что постбайтовая адресация является весьма универсальной и позволяет адресовать как общие регистры, так и ячейки памяти с указанием любого варианта вычисления эффективного адреса *EA*. Однако при адресации только регистров или аккумулятора постбайт оказывается излишним, если трехбитовое поле *reg* разместить в первом байте команды или использовать неявную адресацию. Эта возможность реализуется в специальных (укороченных) форматах, которые содержат минимально необходимое число байтов. Специальные форматы предусмотрены для команд, выполняющих следующие часто используемые операции: включение содержимого регистра в стек, извлечение из стека в регистр, передача непосредственных данных в регистр, инкремент и декремент содержимого регистра, обмен содержимым аккумулятора *AX* и регистра, пересылка между аккумулятором и ячейкой памяти с прямой адресацией, вычитание с участием содержимого аккумулятора и непосредственного операнда. В качестве примера на рис. 2.4 приведены стандартный и специальный форматы команды *INC r*. Программа-ассемблер выбирает более короткий формат команды автоматически.

Рис. 2.4. Пример стандартного (а) и специального (б) форматов команды *INC r*

Всего можно выделить девять существенно различных форматов команд. В табл. 2.4 приведены форматы команд, не обязательные байты заключены в круглые скобки, а также указаны варианты форматов, различающихся назначением битов *b₁* первого байта *B1* формата команды, в котором содержится код операции *COP*. Звездочкой помечены специальные форматы. (В данную таблицу не включены редко используемые двухбайтовые команды *AAM* и *AAD*, оба байта которых заняты кодом операции.)

Таблица 2.4

Формат	Вариант формата	Полный список команд
COP		CLC; CMC; STC; CLD; STD; CLI; STI; HLT; WAIT; NOP; RET; IRET; INTO; INT3; POSHF; POPF; LAHF; SAHF; XLAT; CBW; CWD; AAA; DAA; AAS; DAS; LOCK; REP; REPNZ; CMPS; LODS; MOVS; STOS; SCAS; IN ac.DX; OUT DX.ac b2-bo=reg INC r; DEC r; PUSH r; POP r; XCHG AX,r b2-bo=COP PUSH sr; POP sr b4-b3=sr
COP		LEA; LDS; LES
md reg (disp L) (disp H)	r/m bo=w bo=w, b1=d	TEST r,r; TEST r,m; XCHG r,r; XCHG r,m MOV r,r; MOV r,m; MOV m,r; ADD r,r; ADD r,m; ADD m,r; ADC r,r; ADC r,m; ADC m,r; SUB r,r; SUB r,m; SUB m,r; SBB r,r; SBB r,m; SBB m,r; CMP r,r; CMP r,m; CMP m,r; AND r,r; AND r,m; AND m,r; OR r,r; OR r,m; OR m,r; XOR r,r; XOR r,m; XOR m,r;
COP		PUSH r; PUSH m; POP r; POP m; JMP [r];
md COP (disp L) (disp H)	r/m bo=w bo=w, b1=v	CALL [r]; JMP [m]; CALL [m] INC m; DEC m; NEG; NOT; MUL; IMUL; DIV; IDIV RCL; RCR; ROL; ROR; SAL; SAR; SHL; SHR
COP	bo=w	MOV m,d; MOV r,d; AND m,d; AND r,d;
md COP (disp L) (diap H) data L (data H)	r/m bo=w	OR m,d; OR r,d; XOR m,d; XOR r,d; TEST m,d; TEST r,d; CMP m,d; CMP r,d ADD m,d; ADD r,d; ADC m,d; ADC r,d; SUB m,d; SUB r,d; SBB m,d; SBB r,d
COP data L (data H)	I, bo=w	RET d ADD ac,d; ADC ac,d; SUB ac,d; SBB ac,d; CMP ac,d; AND ac,d; OR ac,d; XOR ac,d; TEST ac,d
	b2-bo=reg b3=w	MOV r,d
COP (disp L) (disp H) COP md Osr (disp L) (disp H)	bo=w	JMP displ6; JMP disp8; Jcond disp8; CALL displ6; MOV ac,m; MOV m.ac (при diap L, disp H) MOV sr,m; MOV sr,r; MOV m.sr; MOV r.sr;
COP addr8 COP offset L offset H seg L seg H		IN addr8; OUT addr8 JMP addr; CALL addr (тип FAR)

2.2. Способы адресации

Команды ЦП ВМ86 реализуют весьма разнообразные способы адресации, что упрощает организацию и использование сложных структур данных, а также расширяет возможности отдельных команд и повышает гибкость их применения.

Регистровая адресация. Операнд находится в одном из общих регистров МП или в одном из сегментных регистров. Регистр может быть определен в байте кода операции или в постбайте, в котором выделены 3-битовые поля *reg* и *r/m* (при *md=11*). Команды, оперирующие содержимым регистров, являются наиболее короткими и выполняются за наименьшее время, так как не требуют вычисления ЕА и выполнения цикла шины для обращения к памяти. Для многих команд с регистровой адресацией предусмотрены специальные укороченные форматы.

Непосредственная адресация. Непосредственными операндами являются константы длиной 8 или 16 бит, которые размещаются в последних байтах команды (младший байт следует первым). Доступ к таким операндам в ВМ86 осуществляется очень быстро, поскольку во время выполнения команды они находятся во внутренней очереди команд. Имеются команды, позволяющие манипулировать непосредственными операндами и содержимым общих регистров или ячеек памяти.

Однако отсутствуют команды непосредственной загрузки сегментных регистров и включения константы в стек. Поэтому эти операции осуществляются с помощью промежуточной загрузки общего регистра или ячейки памяти. Однако отсутствуют команды непосредственной загрузки сегментных регистров и включения константы в стек. Поэтому эти операции осуществляются с помощью промежуточной загрузки общего регистра или ячейки памяти.

Прямая адресация является простейшим способом адресации ячейки ЗУ, при котором эффективным адресом ЕА является содержимое байтов смещения *disp* команды. В командах преобразования данных этот способ реализуется при использовании постбайта с полями *md=00* и *r/m=110*. Для пересылок между аккумулятором и памятью предусмотрен укороченный формат.

Разновидностью этого способа является длинная прямая адресация, при которой в формате команды содержатся четыре байта, указывающие базовый адрес сегмента и сегментное смещение *offset*. В этом случае обеспечивается доступ к ячейке с любым логическим адресом, т. е. к произвольной ячейке во всем пространстве адресов 1 Мбайт, но длинная прямая адресация используется только в командах межсегментных переходов и вызовов подпрограмм. Другая разновидность прямой адресации применяется для указания портов ввода - вывода в двухбайтовых командах *IN* и *OUT*, второй байт которых содержит адрес (номер) порта.

Косвенная регистровая адресация. В командах преобразования данных эффективный адрес ЕА равен содержимому одного из регистров *SI*, *DI*, *BX* и *BP* при соответствующем кодировании полей *md* и *reg* постбайта: *md=00*; *r/m=100, 101, 111* и *md=01, r/m=110*; *dispL=0*. В командах безусловного перехода и вызова подпрограммы с регистровой косвенной адресацией допускается указание любого 16-битового общего регистра (при *md=11*; *r/m=000,..., 111*).

Данный способ адресации позволяет вычислять адреса во время выполнения программ, что часто требуется, например, для обращения к различным элементам таблиц данных. При модификации содержимого регистра одна и та же команда оперирует различными ячейками памяти, что позволяет организовать вычислительные циклы. Изменение содержимого регистра обычно осуществляется с помощью команд инкрементирования и декрементирования, а также других арифметических команд и команды загрузки эффективного адреса *LEA*. Разновидностью этого способа является косвенная адресация портов ввода - вывода с помощью содержимого регистра *DX* в однобайтовых командах *IN* и *OUT*.

Базовая адресация. Эффективный адрес операнда ЕА вычисляется путем суммирования содержимого базовых регистров *BX* или *BP* и смещения *disp* (при *md=01, 10* и *r/m=111, 110*). При использовании *BX* происходит обращение к операнду в текущем сегменте данных, а при использовании *BP* - в текущем сегменте стека. Смещения, содержащиеся в команде, могут иметь длину 8 или 16 бит и интерпретируются как знаковые целые, представленные в дополнительном коде.

Базовая адресация обычно используется для подступа к элементам структуры данных, когда смещение (номер) элемента известно на стадии разработки программы (при ее ассемблировании), а базовый (начальный) адрес структуры должен вычисляться при выполнении программы. Модификация содержимого базового регистра позволяет обратиться к одноименному элементу различных структур данных.

Индексная адресация. Значение ЕА вычисляется как сумма смещения *disp*, находящегося в команде, и содержимого индексного регистра *SI* или *DI* (при *md=01, 10* и *r/m=100, 101*). Данный способ обычно применяется для обращения к различным элементам одномерного массива (таблицы) данных, когда смещение определяет известный при ассемблировании начальный адрес массива, а индексный регистр, содержимое которого может модифицироваться при выполнении программы, определяет элемент массива. По существу индексная адресация в ВМ86 аналогична базовой.

Базовая индексная адресация. Эффективный адрес ЕА равен сумме со-держимого базового регистра ВХ или ВР, индексного регистра SI или DI и сме-щения *disp*, находящегося в команде (в частном случае смещение может отсут-ствовать) . Этот способ реализуется при следующем кодировании полей пост-байта: *mdNell*; *r/m=000, 001, 010, 011* и обеспечивает наибольшую гибкость адресации, так как два компонента адреса можно определить и варьировать при выполнении программы. Это удобно при обращении к элементам матриц, т. е. к двумерным массивам, представляемым в памяти как совокупность одномерных массивов.

Относительная адресация ЦП ВМ86 реализуется только по отношению к указателю команд IP, так что сегментное смещение вычисляется как сумма смещения *disp*, находящегося в команде, и текущего значения IP. При этом значение IP равно адресу байта, следующего за рассматриваемой командой, которая в это время выполняется микропроцессором. В ВМ86 относительная адресация не используется в командах, оперирующих данными, а применяется только в командах условных и безусловных переходов, вызовов подпрограмм и управления циклами. Смещение по отношению к содержимому IP не зависит от размещения программ в адресном пространстве памяти, что обеспечивает позиционную независимость команд. При автоматизированном ассемблирова-нии программы указывается метка команды, которой передается управление, а необходимое смещение вычисляется программой-ассемблером.

Неявная адресация. Объект, содержимым которого манипулирует команда, указывается с помощью первого байта команды вместе с кодом операции без выделения специального поля для этой цели. Чаще всего этот специфический способ адресации встречается в однобайтовых командах, где адресуемым объектом являются аккумулятор, регистр флагов или отдельные флаги. В частности, в командах обработки цепочек неявно используются индексные регистры, причем регистр SI адресуется первый элемент цепочки-источника, а регистр DI - первый элемент цепочки-получателя.

2.3. Описание системы команд

Система команд ВМ86 содержит 91 мнемокод и позволяет совершать операции над байтами, двухбайтовыми словами, отдельными битами, а также цепочками байтов и слов. Имеется широкий набор арифметических команд, включающий умножение и деление, который ориентирован на обработку как беззнаковых, так и знаковых чисел. Весьма разнообразны команды пересылки данных, логических операций, переходов в программе и вызовов подпрограмм, а также управления микропроцессором. Число вариантов команд, т.е. число различных машинных кодов, превышает 3800 благодаря использованию восьми способов адресации в их различных модификациях. Тем самым обеспечивается гибкость применения большого числа команд, что позволяет выбрать наиболее рациональные способы адресации в конкретных случаях. Это особенно важно при обработке сложных структур данных.

По функциональному признаку система команд ВМ86 разбивается на шесть групп: пересылка данных, арифметические операции, логические операции и сдвиги, передача управления, обработка цепочек и управление микропро-цессором. В наиболее компактной форме система команд представлена в табл. 2.5, в которой арифметические и логические команды объединены исхо-дя из общности параметров этих команд и одинаковой последовательности действия МП при их выполнении. В табл. 2.5 введены следующие обозначения:

г - общий регистр; *sg* - сегментный регистр; *m* - адрес ячейки памяти, кото-рый указывается в мнемокоде в соответствии с используемым способом адреса-ции; *d* - непосредственный операнд; *ac* - аккумулятор АХ или AL; *p* - адрес 8-разрядного порта ввода - вывода; *disp* - смещение при адресации относител-но IP; *addr* - указатель адреса при межсегментных переходах и вызовах; *ty-pe* - тип (вектор) прерывания; *cond* - условия в команде условных переходов.

В табл. 2.5 представлены все мнемонические обозначения команд и все допустимые варианты представления операндов. Для каждого варианта указа-но число байтов в формате *nB* и число тактов синхронизации *nT*, требуемое для выполнения команды. Значение *E*, равное числу тактов, которое требуется для вычисления эффективного адреса ЕА, следует брать из табл. 2.6 в соответ-ствии с указанным в команде способом адресации. Если имеется два варианта команды, определяющие одно и то же действие, то в табл. 2.5 приводится более короткий вариант. Команды, которые имеют общие форматы и одинаковые варианты представления операндов при одинаковом времени выполнения (т.е. различаются только операцией, выполняемой над операндами), приво-дятся в одной строке.

Влияние команд на флаги иллюстрируют табл. 2.7, 2.8, в которые включены только те команды, которые это влияние оказывают. Знак вопроса соответству-ет случаям, когда состояние флага после выполнения команды произвольно (например, зависит от конкретных значений операндов).

Полный набор команд, упорядоченный по мнемокоду, машинные коды команд и описание выполняемых операций приведены в табл. 2.9.

Для удобства пользования в таблице представлены все варианты мнемони-ческих обозначений команд условных переходов и команд управления циклами. В дополнение к введенным ранее обозначениям в данной таблице используются следующие: disp 8/16-смещение в формате команды, состоящее из одного или двух байтов; d 8/16 - одно- или двухбайтовая константа; sbr - имя под-программы; diff- разница между адресом перехода и содержимым указателя

Таблица 2.5

Мнемо-код	Опе-ранд	пв	пг	Мнемо-код	Опе-ранд	пв	пг	Мнемо-код	Опе-ранд	пв	пг
MOV	r, r	2	2					LOOPZ	diap	2	18/6
	r.m	2-4	8+E	Corr ³⁾	-	1	4	LOOPNZ			
	m,r	2-4	9+E					JCXXZ			
	ac,m	3	11	Flag ⁴⁾	-	1	2	INT	-	1	52
	m,,ac	3	11						type	2	51
	r,d	2-3	4	MUL ⁵⁾	r/m	2-4	133	INTO	-	1	53/4
	m.d	3-6	10+E								
	r, ar	2	2	IMUL ⁵⁾	r/m	2-4	154	IRET	-	1	24
	s,r,r	2	2					REP	-	1	6
	m, sr	2-4	9+E	DIV ⁵⁾	r/m	2-4	162				
	ar,m	2-4	8+E	IDIV ⁵⁾	r/m	2-4	184				
PUSH	r	1	10					MOVS	-	1	18
	m	2-4	16+E	AAM	-	2	83	CMPS ⁸⁾	-	2	9+13
	sr	1	10	AAD	-	2	60	SCAS ⁸⁾	-	1	15
PUSHF	-	1	10	SBW	-	1	5		-	2	9+15
POP	r	1	8	CWD	-	1	5				
	m	2-4	17+E					LODS ⁸⁾	-	1	12
	ar	1	8	SHL	r	2	2			2	9+13
				SAL	m	2-4	15+E				
POPF	-	1	8	SHL	r,CL	2	8+4N	STOS ⁸⁾	-	1	11
				SAR	m,CL	2-4	20+E+4N			2	9+10
XCHG	r, r	2	4								
				ROL				WAIT	-	1	3
				ROR							
XLAT	-	1	11	RCL				NOP	-	1	3
				RCR							
IN	p.ac	2	10					HLT	-	1	2
	ac,[DX]	1	8	JMP ⁶⁾	disp	2-3	15				
					r	2	11	ESC		r 2	2
OUT	p.ac	2	10		m	2-4	18+E			m 2-4	8+E
	[DX],ac	1	8								
LEA	r,m	3-4	2+E	JMP ⁷⁾	addr	5	15	LOCK	-	1	2
LDS	r,m	3-4	16+E		m	2-4	24+E	TEST ¹⁾			
LES	r,m	3-4	16+E	Jcond	disp	2	8/4				
INC	r	1	2	CALL ⁶⁾	disp	3	19				
DEC	m	2-4	15+E		r	2	16				
					m	2-4	21+E				
NEG	r	2	3								
NOT	m	2-4	16+E	CALL ⁷⁾	addr	5	28				
					m	4	37+E				
ADD	r,r	2	3								
ADC	r,m	2-4	9+E	RET ⁶⁾	-	1	B				
SUB	m,r ²⁾	2-4	16+E		d	3	12				
SBB	r,d	3-4	4								
CMP	ac,d	2-3	4	RET ⁷⁾	-	1	12				
AND ¹⁾	m,d	3-6	17+E		d	3	18				
OR ¹⁾											
XOR ¹⁾				LOOP	disp	2	17/5				

Примечание.

1) Логические команды (в отличие от приведенных в этой же строке арифметических команд) не имеют формата, в котором при работе со словами (т.е. при w = 1) указывается однобайтовый непосредственный операнд d.

- 2) Команда TEST не имеет варианта m, r со временем выполнения 16+E, так как он был бы эквивалентен более короткому варианту r, m.
- 3) Cогг - команды коррекции при сложении и вычитании: DAA, DAS, AAA, AAS.
- 4) Flag используется для обозначения команд управления флагами CLC, STC, CMC, CLI, STI, CLD, STD.
- 5) Для команд умножения и деления указано время выполнения при работе с операндами максимальной допустимой длины, размещенными в регистрах.
- 6) Команда осуществляет внутрисегментный переход, вызов или возврат.
- 7) Команда осуществляет межсегментный переход, вызов или возврат.
- 8) Дано время выполнения одного цикла цепочечной команды, имеющей префикс повторения команд IP (при адресации относительно IP); label — метка, к которой осуществляется переход. Для каждой цепочечной команды приведено только общее mnemonicское обозначение, которое требует дополнительного указания порядности элементов цепочки, например MOVSB — пересылка байтов, MOVSW — пересылка слов.

Данные табл. 2.9 позволяют осуществить переход от мнемонических кодов к машинным кодам команд. При необходимости выполнить обратный переход целесообразно воспользоваться системой команд, упорядоченной по машинному коду операции. Естественно, что наиболее удобным является автоматическое преобразование кодов с помощью соответствующих программ ассемблера и ре-ассемблера

Таблица 2.6.

Адресация	Способ вычисления ЕА	Обозначение в мнемоническом коде	Число тактов Е (ЕА)
Прямая	disp	disp	6
Косвенная	[BX], [BP], [SI], [DI]	[r]	5
Базовая или индексная	[BX, BP, SI, DI] + disp	[r]disp	9
Базовая индексная:			
Без смещения	[BP + DI], [BX + SI]	[r][r]	7
	[BP + SI], [BX + DI]		8
со смещением	[BP + DI] + disp	[r]disp[r]	11
	[BX + SI] + disp		11
	[BP + SI] + disp		12
	[BX + DI] + disp		12

Таблица 2.7

Операции	Команды	Флаги состояний					
		OF	CF	AF	SF	ZF	PF
Сложение, вычитание	ADD ADC SUB SBC	+	+	+	+	+	+
	CMP NEG CMPPB SCAS	+	+	+	+	+	+
	NC DEC	+	-	+	+	+	+
Умножение	MUL IMUL	+	+	?	?	?	?
Деление	DIV ID IV	?	?	?	?	?	?
Десятичная коррекция	DAA DAS	?	+	+	+	+	+
	AAA AAS	?	+	+	?	?	?
	AAM AAD	?	?	?	+	+	+
Логические	AND OR XOR TEST	0	0	?	+	+	+
Сдвиги	SHL SHR ¹⁾	+	+	?	+	+	+
	SHL SHR ²⁾	?	+	?	+	+	+
	SAR	0	+	?	+	+	+
	ROL ROR RCL RCR ¹⁾	+	+	-	-	-	-
	ROL ROR RCL RCR ²⁾	?	+	-	-	-	-
Восстановление флагов	POPF IRET	+	+	+	+	+	+
	SAHF	-	+	+	+	+	+
Управление флагом	STC	-	1	-	-	-	-
	CLC	-	0	-	-	-	-
переноса	CMC	-	r	-	-	-	-

Примечание. «+» - результат операции влияет на флаг; «-» - не влияет; 1 - устанавливает в «1»; 0 - устанавливает в «0»; r - инвертирует;

? - не определен.

1) Одиночный сдвиг.

2) Многобайтный сдвиг.

Таблица 2.8

Операции		Команды	Флаги управления		
			DF	IF	TF
Восстановление флагов	POPF	IRET	+	+	+
Прерывания	INT	INTO	-	0	0
Управление флагами	STD		1	-	-
	CLD		0	-	-
	STI		-	1	-
	CLI		-	0	-

Примечание. «+» - результат операции влияет на флаг; «-» - не влияет; 1 - устанавливает в «1»; 0 - устанавливает в «0»; r - инвертирует;

? - не определен.

¹⁾ Одиночный сдвиг.

²⁾ Многоразрядный сдвиг.

Таблица 2.9

N п/п	Команда	Байты кода команды					Символическое описание содержание операции	и/или		
		B1	B2	B3-B6						
1	AAA	00110111						ASCII - коррекция для сложения		
2	AAD	11010101	00001010						ASCII - коррекция для деления	
3	AAM	11010100	00001010						ASCII - коррекция для умножения	
4	AAS	00111111						ASCII - коррекция для вычитания		
5	ADC	r, r/m	0001001W	md	reg	r/m	(disp8/16)	r←r/r/m+CF; сложение с переносом		
		r/m, r	0001000W	md	reg	r/m	(disp8/16)	r/m←r/r/m+CF		
		r/m, d	1000000W	md	010	r/m	(disp8/16)d8/16	r/m←r/m+d+CF		
		rb6/ml6,d8	10000011	md	010	r/m	data L	r/m←r/m+d+CF		
		ac, d	0001010W	data L			(data H)		ac←ac+d+CF	
6	ADD	r, r/m	0000001W	md	reg	r/m	(disp8/16)	r←r+r/m; сложение		
		r/m, r	0000000W	md	reg	r/m	(disp8/16)	r/m←r+r/m		
		r/m, d	1000000W	md	000	r/m	(disp8/16)d8/16	r/m←r/m+d		
		rb6/ml6,d8	10000011	md	000	r/m	data L	r/m←r/m+d		
		ac, d	0000010W	data L			(data H)		ac←ac+d	
7	AND	r, r/m	0010001W	md	reg	r/m	(disp8/16)	r←r∧r/m; конъюнкция, И		
		r/m, r	0010000W	md	reg	r/m	(disp8/16)	r/r←r∧r/m		
		r/m, d	1000000W	md	100	r/m	data L data H	r/m←r/m∧d		
		ac, d	0010010W	data L			(data H)		ac←ac∧d	
8	CALL	NEAR sbr	11101000	diff	L	diff H		IP←IP+diff; вызов прямой		
		NEAR rb6/ml6	11111111	md	010	r/m	(disp8/16)	IP←IP+r/m; вызов косвенный		
		FAR sbr	10011010	IP-L			IP-H CS-L CS-H		IP←IP-L, IP-H; CS←CS-L, CS-H; вызов прямой	
		FAR m32	11111111	md	011	r/m	(disp8/16)	IP←ml6;CS←ml6; вызов косвенный		
9	CBW	10011000						Преобразование байта в слово		
10	CLC	11111000						CF←0; сброс переноса		
11	CLD	11111100						DF←0; сброс триггера направления		
12	CLI	11111010						IF←0; запрет прерыва- ния		
13	CMC	11110101						CF←CF; инверсия пере носа		
14	CMP	r,r/m	0011101W	md	reg	r/m	(disp8/16)	r-r/m; сравнение		
		r/m,r	0011100W	md	reg	r/m	(disp8/16)	r/m-r		
		r/m, d	1000000W	md	111	r/m	(disp8/16)d8/16	r/m-d		
		rb6/ml6,d8	10000011	md	111	r/m	data L	r/m-d		
		ac, d	0011110W	data L			(data H)		ac-d	

Продолжение Таблицы 2.9

N п/п	Команда	Байты кода команды			Символическое описание и/или содержание операции
		B1	B2	B3-B6	
15	CMPS	1010011W			Сравнение элементов цепочек
16	CTO	10011001			Преобразование слова в двойное слово
17	DAA	00100111			Десятичная коррекция для сложения
18	DAS	00101111			Десятичная коррекция для вычитания
19	DEC r/m r	1111111W 01001reg	md 001 r/m (disp8/16)		$r/m \leftarrow r/m - 1$; декремент
20	DIV r/m	1111011W	md 110 r/m (disp8/16)		Деление чисел без знака
21	ESC OP CODE, r/m	11011XXX	md YYY r/m (disp8/16)		Переключение на сопроцессор
22	HLT	11110100			Останов
23	IDIV r/m	1111011W	md 111 r/m (disp8/16)		Деление чисел со знаком
24	IMUL r/m	1111011W	md 101 r/m (disp8/16)		Умножение чисел со знаком
25	IN ac.port. ac.DX	1110010W 1110110W	port8		Ввод из фиксированного порта Ввод из порта с косвенной адресацией
26	INC r/m r	1111111W 01000reg	md 000 r/m (disp8/16)		$r/m \leftarrow r/m + 1$; инкремент $r \leftarrow r + 1$;
27	INT type INT0 INT3	11001101 11001110 11001100			Прерывание заданного типа Прерывание по переполнению Прерывание типа 3
28	IRET	11001111			Возврат из прерывания
29	JA label	01110111	diff L		$IP \leftarrow IP + \text{diff L}$; перейти, если выше
30	JAЕ label	01110011	diff L		$IP \leftarrow IP + \text{diff L}$; если выше или равно
31	JB label	01110010	diff L		$IP \leftarrow IP + \text{diff L}$; если ниже
32	JBE label	01110110	diff L		$IP \leftarrow IP + \text{diff L}$; если ниже или равно
33	JC label	01110010	diff L		$IP \leftarrow IP + \text{diff L}$; если есть перенос
34	JCXZ label	11100011	diff L		$IP \leftarrow IP + \text{diff L}$; если CX равен нулю
35	JE label	01110100	diff L		$IP \leftarrow IP + \text{diff L}$; если равно
36	JG label	01111111	diff L		$IP \leftarrow IP + \text{diff L}$; если больше
37	JGE label	01111101	diff L		$IP \leftarrow IP + \text{diff L}$; если больше или равно
38	JL label	01111100	diff L		$IP \leftarrow IP + \text{diff L}$; если меньше
39	JLE label	01111110	diff L		$IP \leftarrow IP + \text{diff L}$; если меньше или равно
40	JMP SHORT label NEAR label NEAR r16/m16 FAR label FAR m32	11101011 11101001 11111111 11101010	diff L diff L md 100 r/m (disp8/16) IP-L IP-H CS-L CS-H	diff H	$IP \leftarrow IP + \text{diff L}$; прямой короткий переход $IP \leftarrow IP + \text{diff L}$; прямой переход $IP \leftarrow IP + r/m$; косвенный переход $IP \leftarrow IP-L, IP-H$; $CS \leftarrow CS-L, CS-H$; прямой переход $IP \leftarrow m16$; $CS \leftarrow m16$; косвенный переход
41	JNA label	01110110	diff L		$IP \leftarrow IP + \text{diff L}$; если не выше
42	JNAE label	01110011	diff L		$IP \leftarrow IP + \text{diff L}$; если не выше или равно
43	JNB label	01110111	diff L		$IP \leftarrow IP + \text{diff L}$; если не ниже

Продолжение Таблицы 2.9

N п/п	Команда	Байты кода команды			Символическое описание и/или содержание операции
		B1	B2	B3-B6	
44	JNC label	01110011	diff L		IP←IP+diff L; если нет переноса
45	JNE label	01110101	diff L		IP←IP+diff L; если не равно
46	JNG label	01111110	diff L		IP←IP+diff L; если не больше
47	JNGE label	01111100	diff L		IP←IP+diff L; если не больше или равно
48	JNL label	01111101	diff L		IP←IP+diff L; если не меньше
49	JNLE label	01111111	diff L		IP←IP+diff L; если не меньше или равно
50	JNO label	01110001	diff L		IP←IP+diff L; если нет переполнения
51	JNP label	01111011	diff L		IP←IP+diff L; если нет паритета
52	JNS label	01111001	diff L		IP←IP+diff L; если результат положительный
53	JNZ label	01110101	diff L		IP←IP+diff L; если не ноль
54	JO label	01110000	diff L		IP←IP+diff L; если есть переполнение
55	JP label	01111010	diff L		IP←IP+diff L; если есть паритет
56	JPE label	01111010	diff L		IP←IP+diff L; если паритет четный
57	JPO label	01111011	diff L		IP←IP+diff L; если паритет нечетный
58	JS label	01111000	diff L		IP←IP+diff L; если результат отрицательный
59	JZ label	01110100	diff L		IP←IP+diff L; если ноль
60	LAHF	10011111			AH←FL; пересылка младшего байта F в AH
61	LDS r16,m32	11000101	md reg r/m (disp8/16)		r, DS←m32; загрузка указателя адреса
62	LEA r16,m	10001101	md reg r/m (disp8/16)		r←EA; загрузка эффективного адреса
63	LES r16,m32	11000100	md reg r/m (disp8/16)		r, ES←m32; загрузка указателя адреса
64	LODS	1010110W			Загрузка элемента цепочки
65	LOOP label	11100010	diff L		IP←IP+diff L; зациклить
66	LOOPE/LOOPZ				
	label	11100001	diff L		IP←IP+diff L; зациклить, если равно
67	LOOPNE/LOOPNZ				
	label	11100000	diff L		IP←IP+diff L; зациклить, если не равно
68	MOV r, r/m	1000101W	md reg r/m (disp8/16)		r←r/m; пересылка
	r/m, r	1000100W	md reg r/m (disp8/16)		r/m←r;
	r/m, d	1100011W	md 000 r/m (disp8/16)d8/16		r/m←d;
	r, d	1011Wreg	data L	(data H)	r←d
	ac, m	1010000W	disp L	disp H	ac←m; при прямой адресации
	m, ac	1010001W	disp L	disp H	m←ac; при прямой адресации
	sr, r/m	10001110	md Osr r/m (disp8/16)		sr←r/m; запись в сегментный регистр
	r/m, sr	10001100	md Osr r/m (disp8/16)		r/m←sr; чтение из сегментного регистра
69	MOVS	1010010W			Пересылка элемента цепочки
70	MUL r/m	1111011W	md 100 r/m (disp8/16)		Умножение чисел без знака
71	NEG r/m	1111011W	md 011 r/m (disp8/16)		Смена знака числа
72	NOP	10010000			Пустая операция
73	OR r, r/m	0000101W	md reg r/m (disp8/16)		r←r r/m; дизъюнкция, ИЛИ

Продолжение Таблицы 2.9

N п/п	Команда	Байты кода команды			Символическое описание и/или содержание операции
		B1	B2	B3-B6	
	r/m, r	0000100W	md	reg r/m (disp8/16)	$r/m \leftarrow r/m \vee r$
	r/m, d	1000000W	md	001 r/m (disp8/16)d8/16	$r/m \leftarrow r/m \vee d$
	ac, d	0000110W	data L	(data H)	$ac \leftarrow ac \vee d$
74	OUT ac, port	1110011W	port8		Вывод в фиксированный порт
	ac, DX	1110111W			Вывод в порт при косвенной адресации
75	POP r/m	10001111	md	000 r/m (disp8/16)	Чтение из стека
	r	01011reg			
	sr	000sr111			
76	POPF	10011101			Загрузка регистра флагов из стека
77	PUSH r/m	11111111	md	110 r/m (disp8/16)	Запись в стек
	r	01010reg			
	sr	000sr110			
78	PIUSHF	10011100			Запись регистра флагов в стек
79	RCL r/m, 1	1101000W	md	010 r/m (disp8/16)	Циклический сдвиг влево через CF
	r/m, CL	1101001W	md	010 r/m (disp8/16)	Циклический сдвиг влево через CF
80	RCR r/m, 1	1101000W	md	011 r/m (disp8/16)	Циклический сдвиг вправо через CF
	r/m, CL	1101001W	md	011 r/m (disp8/16)	Циклический сдвиг вправо через CF
81	RET (NEAR)	11000011			Возврат из подпрограммы
	d(NEAR)	11000010	data L	(data H)	Возврат с прибавлением константы к SP
	r(NEAR)	11001011			Возврат с прибавлением константы к SP
	d(FAR)	11001010	data L	(data H)	Возврат с прибавлением константы к SP
82	ROL r/m, 1	1101000W	md	000 r/m (disp8/16)	Циклический сдвиг влево
	r/m, CL	1101001W	md	000 r/m (disp8/16)	Циклический сдвиг влево
83	ROR r/m, 1	1101000W	md	001 r/m (disp8/16)	Циклический сдвиг вправо
	r/m, CL	1101001W	md	001 r/m (disp8/16)	Циклический сдвиг вправо
84	SAHF	10011110			Переноска AH в регистр E
85	SAL r/m, 1	1101000W	md	100 r/m (disp8/16)	Арифметический сдвиг влево
	r/m, CL	1101001W	md	100 r/m (disp8/16)	Арифметический сдвиг влево
86	SAR r/m, 1	1101000W	md	111 r/m (disp8/16)	Арифметический сдвиг вправо
	r/m, CL	1101001W	md	111 r/m (disp8/16)	Арифметический сдвиг вправо
87	SBB r, r/m	0001101W	md	reg r/m (disp8/16)	$r \leftarrow r - r/m - CF$; вычитание с заемом
	r/m, r	0001100W	md	reg r/m (disp8/16)	$r/m \leftarrow r - r/m - CF$
	r/m, d	1000000W	md	011 r/m (disp8/16)d8/16	$r/m \leftarrow r/m - d - CF$
	r16/m16, d8	10000011	md	011 r/m data L	$r/m \leftarrow r/m - d - CF$
	ac, d	0001110W	data L	(data H)	$ac \leftarrow ac - d - CF$
88	SCAS	1010111W			Сканировать элемент цепочки
89	SHL r/m, 1	1101000W	md	100 r/m (disp8/16)	Логический сдвиг влево
	r/m, CL	1101001W	md	100 r/m (disp8/16)	Логический сдвиг влево
90	SHR r/m, 1	1101000W	md	101 r/m (disp8/16)	Логический сдвиг вправо
	r/m, CL	1101001W	md	101 r/m (disp8/16)	Логический сдвиг вправо
91	STC	11111001			$CF \leftarrow 1$; установка флага переноса
92	STD	11111101			$DF \leftarrow 1$; установка триггера направления
93	STI	11111011			$IF \leftarrow 1$; разрешение прерывания
94	STOS	1010101W			$[DI] \leftarrow ac$; запоминание ac в цепочке
95	SUB r, r/m	0010101W	md	reg r/m (disp8/16)	$r \leftarrow r - r/m$; вычитание
	r/m, r	0010100W	md	reg r/m (disp8/16)	$r/m \leftarrow r - r/m$
	r/m, d	1000000W	md	101 r/m (disp8/16)d8/16	$r/m \leftarrow r/m - d$
	r16/m16, d8	10000011	md	101 r/m data L	$r/m \leftarrow r/m - d$
	ac, d	0010110W	data L	(data H)	$ac \leftarrow ac - d$
96	TEST r, r/m	1000010W	md	reg r/m (disp8/16)	$r \wedge r/m$; проверка, неразрушающее И
	m, r	1000010W	md	reg r/m (disp8/16)	$r \wedge r/m$
	r/m, d	1111011W	md	000 r/m (disp8/16)d8/16	$r/m \wedge d$
	ac, d	1010100W	data L	(data H)	$ac \wedge d$
97	WAIT	00111011			Ожидание

Продолжение Таблицы 2.9

В Ком анд а	Байты	кода	команды	Символическое описание и/или содержание операции
	B1	B2	B3-B6	
98	XCHG AX, r	10010reg		AX ↔ r; обмен
	r, AX	10010reg		r ↔ AX
	r, r/m	1000011W	md reg r/m (diap8/16)	r ↔ r/m
	m, r	1000011W	md reg r/m (dis p8/16)	m ↔ r
99	XLAT	11010111		Преобразование байта из AL
100	XOR r, r/m	0011001W	nid reg r/m (diap8/16)	r ← r ⊕ r/m; суммирование по модулю 2 (исключающее ИЛИ)
	r/m, r	0011000W	md reg r/m (dis p8/16)	r/m ← r ⊕ r/m;
	r/m, d	1000000W	md 110 r/m (diap8/16)d8/16	r/m ← r/m ⊕ d
	ac d	0011010W	data L (data H)	ac ← ac ⊕ d
	Префиксы:			
	LOCK	11110000		Префикс блокировки шины
	REP/REPE/REPZ	11110011		Повторять цепочечную операцию
	REPNE/REPNZ	11110010		То же
	SEG	001sr110		Префикс замены сегмента

2.4. Особенности выполнения отдельных команд

Приведем здесь описание лишь некоторых команд, выбранных по принципу частоты их использования или трудности освоения. Описание остальных команд можно найти в [2—5].

Команды пересылки данных составляют четыре подгруппы: общие, обращения к стеку, ввода — вывода и пересылки цепочек (последние рассмотрены вместе с другими цепочечными командами). Эти команды, за исключением POPF и SAHF, не влияют на флаги.

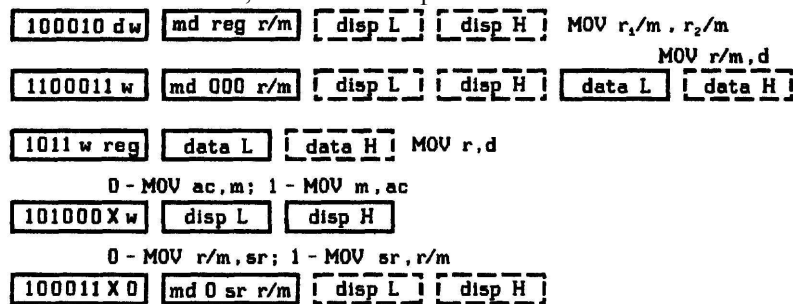


Рис. 2.5. Форматы команды MOV

Команда MOV (рис. 2.5) осуществляет пересылку содержимого источника src в получатель dst и имеет обобщенное представление:

MOV dst, src; dst ← src

Команда MOV r1/m, r2/m содержит постбайт и обеспечивает пересылки регистр - регистр/память и память - регистр при использовании любого общего регистра и любого способа адресации памяти. Бит w определяет передачу байта или слова, а бит d - направление передачи. Команда MOV r/m, d позволяет передать непосредственные данные в общий регистр или ячейку памяти. Команда MOV r, d представляет более короткий вариант (специальный формат) предыдущей команды и осуществляет загрузку констант в общие регистры.

Команды MOV ac, m и MOV m, ac предназначены для загрузки и запоминания содержимого аккумуляторов AL и AX при использовании прямой адресации. Обращение производится к текущему сегменту данных, и адрес, указанный в команде, представляет смещение в этом сегменте. Если пересылаются два байта, то младший располагается по указанному адресу, а старший - по следующему адресу.

Команды MOV sr, r/m и MOV r/m, sr осуществляют пересылки между сегментным регистром и регистром или памятью. При этом передаются только слова, а ячейка памяти может быть определена с помощью любого допустимого способа адресации. Следует учитывать, что в команде MOV sr, r/m нельзя указывать сегментный регистр кода CS, так как при этом результат операции неопределен (выполнение этой команды было бы равносильно специальному безусловному переходу в программе).

Так как не существует команды непосредственной загрузки сегментных регистров, то команда MOV sr, r/m используется для инициализации регистров SS, DS и ES, т. е. для определения соответствующих сегментов памяти. Если, например, в регистр DS необходимо загрузить число 8000, то потребуется две команды:

MOV AX, 8000H; Инициализация регистра DS на 8000

MOV DS, AX;

При этом в качестве промежуточного регистра обычно используется AX, так как команда MOV ac,d короче более общей команды MOV m/r,d.

Возможна также инициализация сегментных регистров из программной памяти при использовании префикса замены сегмента для замены DS на CS при вычислении адреса EA в следующих командах с прямой адресацией:

MOV DS, CS: ADS; Инициализация DS

MOV ES, CS: AES; Инициализация ES

MOV SS, CS: ASS; Инициализация SS

Напомним, что МП ВМ86 обеспечивает защиту процесса инициализации регистров SS и SP, состоящего из двух команд:

MOV SS, CS: ASS

MOV SP, CS: ASP

от прерываний, чтобы гарантировать правильную работу стека. (Операнды ADS, AES, ASS, ASP обозначают адреса ячеек памяти, в которых хранятся базовые адреса соответствующих сегментов.)

1000011 w **md reg r/m** **[disp L]** **[disp H]** XCHG r, r/m

10010 reg XCHG ac, r

Рис. 2.6. Команда XCHG

Команда XCHG (рис. 2.6) осуществляет обмен данными между источни-ком и получателем:

XCHG dst,src; dst«src

и имеет два формата. Общий формат позволяет произвести обмен содержимым любой пары общих регистров, а также обмен между общим регистром и ячей-кой памяти при любом допустимом способе адресации. Укороченный формат осуществляет обмен любого общего регистра и аккумулятора AX. Команда XCHG AX, AX, 16-ричный код которой равен 90, используется как команда пустой операции NOP, обеспечивающая задержку 3Т.

XLAT - однобайтная команда с кодом операции D7 предназначена для быстрого преобразования кодов и заменяет содержимое AL на байт из 256-бай-товой таблицы, начальный (базовый) адрес которой содержится в регистре BX (рис. 2.7). Другими словами, содержимое AL используется как индекс таблицы, находящейся в сегменте данных и адресуемой регистром BX. При выполнении этой команды к содержимому BX прибавляется содержимое AL, а полученный результат используется как смещение относительно DS. Адресуе-мый таким образом байт из памяти пересылается в AL.

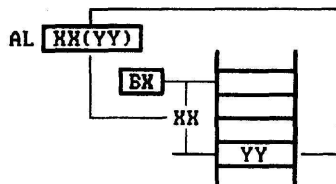


Рис. 2.7. Действие команды XLAT

10001101 **md reg r/m** **[disp L]** **[disp H]** LEA r, m
0 - LES r, m; 1 - LDS r, m
1100010X **md reg r/m** **[disp L]** **[disp H]**

Рис. 2.8. Команды LEA, LDS, LES

Команды LEA, LDS, LES (рис. 2.8) отличаются от других команд пересылки тем, что при их выполнении в адресуемый регистр (регистры) передаются не собственно данные из памяти, а их адреса. Основное назначение этих команд - инициализация регистров перед выполнением цепочечных команд или перед вызовом подпрограммы.

Команда LEA r, m обеспечивает вычисление эффективного адреса EA ячейки памяти в соответствии с указанным способом адресации и загрузку EA (а не содержимого адресуемой ячейки памяти!) в указанный общий регистр. Такая операция может потребоваться, например, для загрузки начального адреса таблицы в регистр BX перед выполнением команды XLAT.

Команды LDS r/m и LES r,m загружают адресную информацию из памяти в адресуемый общий регистр и в соответствующий сегментный регистр. Сначала вычисляется адрес EA памяти, который, как обычно, суммируется с содержимым регистра DS, затем слово из памяти по вычисленному адресу загружается в общий регистр, а следующее слово - в регистр DS (команда LDS) или ES (команда LES). При этом кодирование поля md=11 (регистровая адресация) не должно использоваться, так как действия команд в этом случае не определены.

Команды LDS и LES упрощают коммутацию сегментов данных, поскольку одновременно загружают базовый или индексный регистр и сегментный регистр. Если эти команды подготавливают обращение к цепочке, то в команде LDS указывается регистр DI (цепочка-источник расположена в сегменте данных и адресуется регистром DI), а в команде LES-регистр DI (цепочка-получатель обязательно находится в дополнительном сегменте и адресуется регистром DI).

Арифметические команды выполняются над целыми числами четырех типов: беззнаковыми и знаковыми двоичными, упакованными и неупакованными десятичными.

Беззнаковые двоичные числа могут иметь длину 8 или 16 бит, каждый из которых является значащим, т. е. учитывается при определении значения числа. Это обеспечивает диапазон представления чисел 0-255 и 0- 65 535 соответственно. Для таких чисел имеются команды сложения, вычитания, умножения и деления.

Знаковые двоичные числа также могут содержать 8 или 16 бит, но значащими являются все биты, кроме старшего, который определяет знак числа: 0 - положительное число, 1 - отрицательное число. Соответственно диапазоны значений чисел: от -128 до +128 и от -32 768 до +32 767. Число нуль содержит нули во всех разрядах и считается положительным и четным. Число, имеющее нули во всех разрядах, кроме знакового, равно-27 для 8-битовых чисел и -215 для 16-битовых.

Числа представляются в стандартном дополнительном коде, благодаря применению которого сложение и вычитание как знаковых, так и беззнаковых чисел выполняется с помощью одних и тех же команд (рис. 2.9). Для умножения и деления знаковых чисел предусмотрены специальные команды.

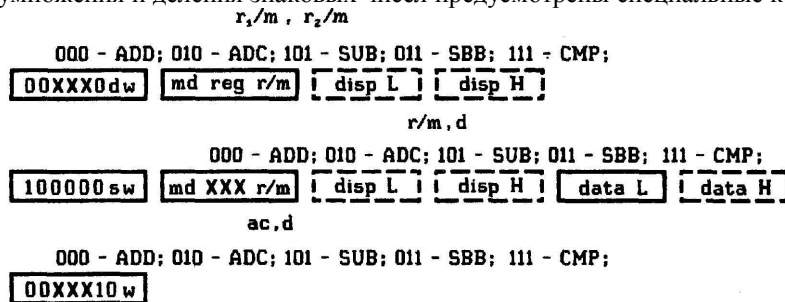


Рис. 2.9. Команды сложения и вычитания

Команда NEG изменяет знак числа, находящегося в общем регистре или ячейке памяти, т. е. формирует его дополнительный код:

NEG dst; dst←0-dst

Например, если однобайтовый операнд равен —1 (1111111), то команда NEG изменит его на +1 (00000001). Если операнд равен нулю, его значение не изменяется. Попытка изменить знак числа — 128 (10000000) или —2¹⁶ не модифицирует операнд, но устанавливает флаг переполнения OF. Отметим, что команда NEG отсутствует в ВМ80, поскольку в нем не предусмотрено специальных средств по обработке знаковых чисел.

Команды умножения и деления (рис. 2.10). В МП ВМ86 имеется по две команды умножения (**MUL** и **IMUL**) и деления (**DIV** и **IDIV**), выполняющие операции с беззнаковыми и знаковыми числами (в дополнительном коде) соответственно. Работа с десятичными числами требует использования специальных команд коррекции AAM и AAD.

Команды умножения

MUL (IMUL) src; ext:ac←acsrc

выполняют умножение адресуемого операнда (общего регистра или ячейку памяти) и содержимого аккумулятора ac. При работе с байтами функции аккумулятора ac выполняет регистр AL, а функции его расширения (ext) — регистр AH, так что 16-битовое произведение формируется в регистре AX. Если перемножаются слова, то множимое располагается в регистре AX, функции расширения которого выполняет регистр DX, так что 32-битовый результат образуется в регистрах DX и AX.

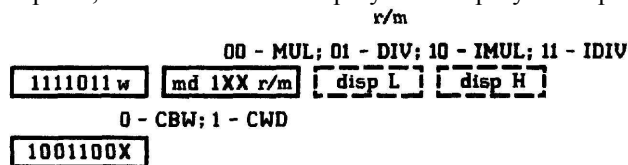


Рис. 2.10. Команды умножения и деления

Команды умножения беззнаковых и знаковых чисел осуществляют практически одни и те же действия и несколько различаются только влиянием на флаги OF и CF (значения остальных арифметических флагов после выполнения этих команд не определены, т. е. эти флаги могут принимать произвольные состояния). При выполнении команды MUL флаги OF и CF устанавливаются в единицу, если старшая половина произведения, находящаяся в регистрах AH или DX, отличается от нулевой, т. е. разрядность результата действительно превышает разрядность операндов. В противном случае эти флаги принимают нулевые значения. При выполнении команды IMUL флаги OF и CF устанавливаются в единицу, если старшая половина разрядов произведения не является расширением знака младшей половины, т. е. не содержит 00 (0000) или FF (FFFF) при умножении байтов (слов). Это означает, что в старшей половине находятся значащие цифры результата. В противном случае OF=CF=0 и старшую половину разрядов произведения можно не сохранять. Отсюда следует, что флаги OF и CF несут однотипную информацию о результатах выполнения команд MUL и IMUL.

Команды деления

DIV (IDIV) src; $ac \leftarrow \text{quot}(\text{ext}:ac/src)$
 $ext \leftarrow \text{rem}(\text{ext}:ac/src)$

производят деление содержимого аккумулятора и его расширения (AH:AL для 8-битового и DX:AX для 16-битового делителя) на содержимое регистра или ячейки памяти src. Частное quot формируется в регистре AL или AX, а остаток rem — в регистре AH или DX. Дробное частное округляется до целого путем отбрасывания дробной части результата. Состояния всех флагов не определены.

Если частное выходит за диапазоны представления чисел в байте (слове) или делитель является нулем, то автоматически генерируется прерывание по ошибке деления (тип 0), а частное и остаток не определены. В результате этого МП переходит к подпрограмме обработки прерывания, полный адрес которой CS:IP берется из ячеек 0000 и 0002.

В ряде случаев, например для предотвращения прерывания, может появиться необходимость до выполнения операции деления проверить возможность возникновения прерывания типа 0. Такая проверка осуществляется с помощью команд:

CMP ext,src; сравнение и переход
JNB OVERFLOW; по переполнению

осуществляющих сравнение старшей половины разрядов делимого с делителем и передающих управление по метке OVERFLOW, если левый операнд в команде CMP больше или равен правому (что является условием возникновения переполнения при делении).

Частное и остаток после выполнения команды IDIV всегда имеют одинаковые знаки. Например, при делении числа —47 на +3 из двух возможных результатов: —15 с остатком —2 и —16 с остатком +1 будет сформирован первый результат.

Однобайтовые команды преобразования разрядности операнда CBW и CWD примыкают к командам деления и осуществляют расширение со знаком операнда, который будет использоваться в качестве делимого. Обе команды не влияют на флаги и не изменяют значения операнда. Команда CBW (код операции 98) реализует преобразование байта в слово путем расширения (копирования) знака содержимого регистра AL в регистр AH. Команда CWD (код операции 99) осуществляет аналогичное преобразование слова в двойное слово путем передачи знака содержимого AX во все биты регистра DX.

Команды CBW и CWD удобно использовать для превращения делимого одинарной длины в делимое двойной длины, что может потребоваться для корректного выполнения команды IDIV. Выполнение команды CBW перед командой IDIV позволяет осуществлять деление 8-битовых чисел, а выполнение команды CWD — деление 16-битовых чисел.

Алгоритмы умножения и деления в МП ВМ86 реализованы не аппаратно, а в виде микропрограмм. Поэтому длительность команд MUL, IMUL, DIV, IDIV включает большое число тактов, причем длительность каждой команды зависит не только от разрядности операндов и расположения операнда src (в регистре или в памяти), но и от конкретных значений операндов в пределах, указанных в табл. 2.10 диапазонов длительности команд. В первом столбце в скобках указана длина сомножителей команд умножения и длина делителя (частного) для команд деления.

Таблица 2.10.

Тип операции (длина операнда)		MUL	IMUL	DIV	IDIV
r-r	(байт)	70–77	80–98	80–90	101–112
r-r	(слово)	118–133	128–154	144–162	165–184
r-m	(байт)	{76–83} +EA	{84–104} +EA	{86–96} +EA	{107–118} +EA
r-m	(слово)	{124–139} +EA	{134–160} +EA	{150–168} +EA	{171–190} +EA

К логическим командам относятся следующие (рис. 2. 11): **AND** (конъюнкция, И), **OR** (дизъюнкция, ИЛИ), **XOR** (исключающие ИЛИ, сумма по модулю два), **TEST** (неразрушающая проверка, которая выполняет конъюнкцию операндов без изменения их значений, но с влиянием на флаги) и **NOT** (инверсия, НЕ). Все логические операции выполняются поразрядно, т. е. каждый бит операндов обрабатывается независимо от других. Для этого в АЛУ микропроцессора включены наборы не связанных друг с другом логических элементов, содержащие по 16 двухвходовых элементов, И, ИЛИ и сумматоров по модулю два, а также набор из 16 инверторов.

Команды передачи управления используются в разветвляющихся и циклических программах, а также при вызове подпрограмм и возврате из них.

Сегментная организация программной памяти определяет два основных типа команд передачи управления: внутрисегментные **NEAR** (близкие) и межсегментные **FAR** (далекие). При выполнении команд типа **NEAR** модифицируется только регистр IP, и адрес переходов представляется одним словом или даже байтом, если используется короткий вариант перехода с ограниченным диапазоном адресов. При выполнении команд типа **FAR** изменяется содержимое регистров IP и CS и адрес перехода представляется двумя словами (сегмент:смещение), что позволяет перейти в любую точку адресного пространства памяти.

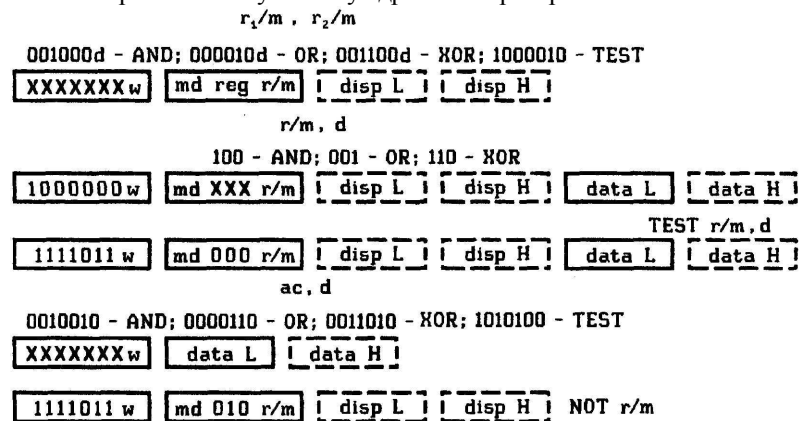


Рис. 5.11. Команды логических операций

В рассматриваемую группу команд входят команды безусловных переходов, вызовов, возвратов, условных переходов, управления циклами и прерываний.

Команды безусловных переходов JMP (рис. 2.12) производят модификацию регистра IP или регистров IP и CS без сохранения прежних значений этих регистров. Имеется три формата (рис. 2.13) команды **JMP** типа **NEAR**, осуществляющих переход в пределах текущего кодового сегмента (внутрисегментный, близкий переход), и два формата команды **JMP** типа **FAR**, осуществляющих переход в любую точку адресного пространства (межсегментный, дальний переход)

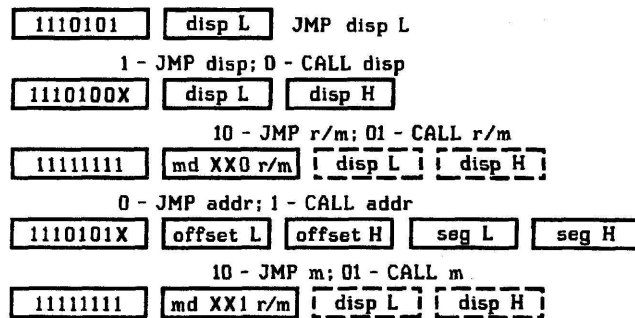
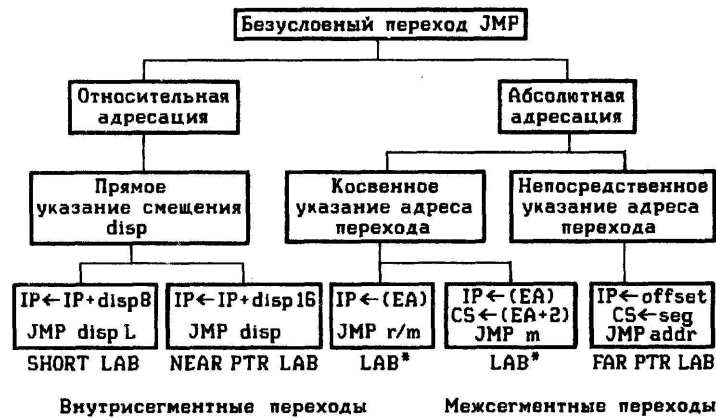


Рис. 2.12. Форматы команд **JMP** и **CALL**

Двухбайтовая команда **JMP disp L** во втором байте содержит смещение которое интерпретируется как знаковое целое. Это смещение добавляется (с предварительным расширением знака до 16 бит) к содержимому IP, которое соответствует адресу команды, находящейся после данной команды **JMP**. Диапазон значений **disp L** составляет от —128 до +127, причем при положительном смещении осуществляется переход вперед, а при отрицательном — переход назад. Данная команда реализует так называемый короткий переход который в мнемоническом обозначении отмечается указателем **SHORT** (напри мер, **JMP SHORT COUNT** — короткий переход к метке **COUNT**). Она находит широкое применение, поскольку большинство переходов в прикладных программах осуществляется на небольшие расстояния.



Трехбайтовая команда **JMP disp** производит такое же действие, как и предыдущая команда, но содержит 16-битовое смещение *disp* (ассемблерный указатель NEAR PTR). Это смещение также интерпретируется как знаковое целое, принимающее значения от $-32\,768$ до $+32\,767$, что обеспечивает переход в любую точку текущего кодового сегмента. Напомним, что благодаря игнорированию переноса из бита 16, возникающего при сложении IP и *disp*, текущий сегмент рассматривается как кольцо. Поэтому диапазон переходов составляет от $-IP$ до $216-IP$, а не от -215 до 215 —!. Если, например, $IP = 8000$ и смещение равно $7FFF$ или меньше, оно вызовет переход вперед, а смещение, равное 8000 или больше, вызовет переход назад.

Первые два рассмотренных формата команды **JMP** реализуют относительный способ адресации, который обеспечивает позиционную независимость программ. Следующие три формата определяют абсолютные адреса переходов и загружают регистр IP (а при межсегментных переходах и регистр CS) новыми значениями, не зависящими от прежних.

Команда **JMP r/m** осуществляет косвенный внутрисегментный переход (ассемблерный указатель WORD PTR), при котором в регистр IP загружается содержимое регистра или ячейки памяти в соответствии с постбайтовым режимом адресации. Необязательные байты *disp L*, *disp H* в команде стандартно относятся к режиму адресации памяти.

Команда **JMP m** реализует косвенный межсегментный переход (ассемблерный указатель DWORD PTR) через содержимое двух ячеек памяти: слово из адресуемой с помощью постбайта ячейки загружается в IP, а слово из следующей ячейки — в регистр CS. Если в постбайте указывается регистр (*rnd=11*), то операция не определена.

Команда **JMP addr** реализует прямой межсегментный переход (ассемблерный указатель FAR PTR) и содержит 4 байта адреса перехода: два байта сегментного смещения *offset* загружаются в регистр IP, а два байта *seg* — в регистр CS.

Команды вызова подпрограммы CALL (см. рис. 2.12) имеют такие же форматы, как и команды **JMP** (кроме команды короткого перехода) и выполняются аналогичным образом, за исключением того, что автоматически запоминается адрес возврата (т. е. адрес команды, следующей за командой **CALL**). С этой целью при внутрисегментных вызовах в стеке запоминается содержимое IP, а при межсегментных вызовах — сначала содержимое IP, а затем CS. Напомним, что включение в стек каждого слова сопровождается уменьшением содержимого SP на два и что SP адресует последнюю заполненную 16-битовую ячейку стека. Использование стека для временного хранения адресов возврата позволяет легко реализовать правильный порядок возвратов из вложенных подпрограмм.

Перечислим действия, выполняемые 5-байтовой командой прямого межсегментного вызова **CALL addr**:

- содержимое регистра SP уменьшается на два;
- в адресуемую регистрами SP и SS ячейку памяти пересылается содержимое CS;
- содержимое SP уменьшается на два;
- в адресуемую регистрами SP и SS ячейку памяти засылается содержимое регистра IP;
- в IP загружается смещение *offset*;
- в CS загружается сегментный адрес *seg*.

В микропроцессоре **BM86** отсутствуют команды условных вызовов подпрограмм (в отличие от **BM80**) и поэтому при необходимости условный вызов реализуется двумя командами. Например, если требуется вызвать подпрограмму **SUBR** при ненулевом результате операции, т. е. реализовать отсутствующую команду **CNZ**, то выполняется команда условного перехода, проверяющая противоположное условие, и осуществляется “перескок” через команду вызова **CALL**:

```

JZ SKIP
CALL SUBR
SKIP: ...

```

В зависимости от того, как определен тип подпрограммы SUBR, ассемблер сформирует необходимую команду внутрисегментного вызова.

Команды возвратов (из подпрограмм) RET (рис. 2.14). Каждая подпрограмма должна содержать хотя бы одну команду возврата RET, которая возвращает управление программе, осуществившей вызов. Такая передача управления осуществляется путем извлечения из стека адреса возврата, включенного в него командой вызова подпрограммы. Поэтому команды возврата не содержат никакой адресной информации и неявно адресуют вершину стека. Тип команды возврата (внутрисегментный NEAR или межсегментный FAR) выбирается в соответствии с типом команды CALL, осуществившей вызов данной подпрограммы.

Однобайтовая команда внутрисегментного возврата RET (код операции C3) выполняет следующие действия: слово из вершины стека загружается в IP, а содержимое SP увеличивается на два.

Однобайтовая команда межсегментного возврата RET (код операции CB) осуществляет следующие действия: слово из вершины стека передается в IP; производится инкремент SP на два; слово из следующей ячейки стека передается в CS; производится инкремент SP на два. В результате в регистрах IP и CS оказывается полный адрес возврата, а SP адресует новую вершину стека.

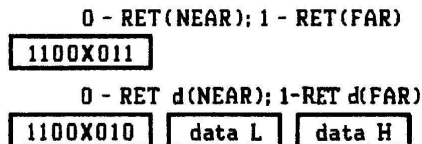


Рис. 2.14. Команды возврата из подпрограмм

Две трехбайтовые команды RET (внутрисегментный возврат с кодом операции C2 и межсегментный возврат с кодом CA) содержат два байта данных, интерпретируемых как беззнаковое целое. Они производят такие же действия, как и соответствующие однобайтовые команды возврата, но дополнительно прибавляют содержащиеся в них данные к указателю стека SP (после извлечения из стека адреса возврата).

Эти команды упрощают возврат из тех подпрограмм, параметры которых передаются в стеке. Прибавление к SP числа, равного числу байтов в блоке параметров, эквивалентно удалению этого блока из стека.

В микропроцессоре BM86 нет команд условного возврата, и поэтому условный возврат из подпрограмм обеспечивается командой условного перехода противоположного смысла и безусловного возврата, как это делалось при эмуляции команд условного вызова:

```
JZ SKIP
RET
SKIP: .....
```

Команды условных переходов (рис. 2.15). Имеется 18 команд условных переходов, которые представлены единым двухбайтовым форматом, позволяющим осуществлять короткие (в пределах от — 128 до +127) переходы относительно указателя команд IP. При выполнении этих команд анализируется некоторое условие, соответствующее текущим состояниям отдельных флагов (кроме флага AF) или некоторым комбинациям флагов (табл. 2.11). Если условие выполнено, то осуществляется переход и однобайтовое смещение, расширенное со знаком до 16 бит, добавляется к содержимому IP. Если условие не выполнено, выполняется следующая по порядку команда. Время выполнения команд составляет восемь тактов в первом случае и четыре такта во втором.

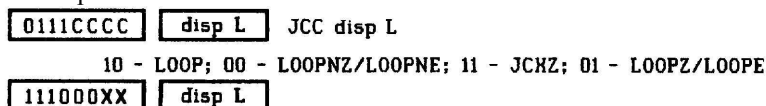


Рис. 2.15. Форматы команд условных переходов и управления циклами

Команды условных переходов обеспечивают ограниченный диапазон переходов, для расширения которого необходимо сочетать их с командами безусловных переходов. Большинство команд условных переходов имеет по два мнемонических обозначения, которые подчеркивают содержательный смысл проверяемого условия и введены для удобства программирования. Обобщенное мнемоническое обозначение команд имеет вид **Jcond label**, где cond соответствует проверяемому условию (первая графа табл. 2.11), двоичный код которого CCCC (вторая графа табл. 2.11) помещается в соответствующее поле первого байта формата команды. Операнд label обозначает команду, к которой осуществляется переход. В качестве label может использоваться метка, а также метка плюс/минус константа или выражение, вычисление которого дает константу.

Команды условных переходов позволяют проверить все отношения между знаковыми и беззнаковыми числами. В мнемониках этих команд при сравнении знаковых чисел для обозначения условия “больше” используется буква G (Greater), а для обозначения “меньше” — буква L (Less). Для аналогичных условий при сравнении беззнаковых чисел используются соответственно буквы A (Above) и B (Below). Условие равенства обозначается буквой E (Equal), а невыполнение некоторого условия — буквой N (Not).

Таблица 2.11

Мнемокод условия	Код CCCC	Условие перехода	
		Логика	Признак
Для чисел со знаком			
G/NLE	1111	Больше/не меньше и не равно	$(SF \oplus OF) \vee ZF = 0$
NG/LE	1110	Не больше/меньше или равно	$(SF \oplus OF) \vee ZF := 1$
L/NGE	1100	Меньше/не больше и не равно	$SA \otimes OF = 1$
NL/GE	1101	Не меньше/больше или равно	$SF \otimes OF = 0$
Для чисел без знака			
A/NBE	0111	Больше/не меньше и не равно	$CF \vee ZF = 0$
NA/BE	0110	Не больше/меньше или равно	$CF \vee ZF = 1$
B/NAE/C	0010	Меньше/не больше и не равно	$CF = 1$
NB/AE/NC	0011	Не меньше/больше или равно	$CF = 0$
Для прочих данных			
E/Z	0100	Равно/по нулю	$ZF = 1$
NE/NZ	0101	Не равно/по не нулю	$ZF = 0$
S	1000	Отрицательный результат	$SF = 1$
NS	1001	Положительный результат	$SF = 0$
O	0000	Есть переполнение	$OF = 1$
NO	0001	Нет переполнения	$OF = 0$
P/PE	1010	Четно	$PF = 1$
NP/PO	1011	Нечетно	$PF = 0$

Первые восемь команд в табл. 2.11 предназначены в основном для использования после команд сравнения CMP, в которых второй операнд вычитается из первого. В результате сравнения двух операндов и последующего перехода в зависимости от состояния флагов реализуются решения, соответствующие операторам отношений $<$, $>$, $\leq$, $\geq$, $\neq$, $=$. Последние восемь команд можно использовать в любой ситуации, когда в зависимости от значения указанного в табл. 2.11 флага следует предпринять одно из двух действий.

Команды условных переходов реализуют относительную адресацию, как и первые два формата команды безусловного перехода JMP. При этом встает задача определения конкретного значения смещения dispL , которое должно быть добавлено к регистру IP для осуществления перехода к указанной метке. Решение этой задачи рассмотрим на примере следующего фрагмента программы:

```
0050 AGAIN: INC CX
0052      ADD AX, [BX]
0054      JNS AGAIN
0056      MOV RESULT, CX
```

В левом столбце приведены эффективные адреса команд, т. е. содержимое IP при выборке каждой команды.

```
—0050
—0056
—6
```

Вычисление показывает, что при ассемблировании команды JNS следует установить смещение $\text{dispL} = \text{FA}$, соответствующее дополнительному коду числа -6.

Команды управления циклами (см. рис. 2.15) используются для удобства реализации вычислительных циклов (итераций). Они осуществляют условный переход в зависимости от состояния регистра CX, который выполняет функции счетчика циклов. Команды LOOP, LOOPE и LOOPNE, кроме того, декрементируют регистр CX перед выполнением условного перехода, что исключает использование соответствующей команды декремента. Как и в командах условных переходов, в данном случае при выполнении условия управление передается команде, расположенной в диапазоне адресов от -128 до +127, задаваемых смещением в байте 2 команды.

Команда LOOP, которая обычно ставится в конце цикла, осуществляет декремент CX - 1 и если $CX \neq 0$, то добавляет смещение dispL к регистру IP; в противном случае ($CX = 0$ и цикл окончен) выполняется следующая по порядку команда. Таким образом, смысл этой команды состоит в повторении цикла CX раз, т. е. в выполнении переходов до тех пор, пока $CX \neq 0$.

Команда JCXZ имеет противоположный смысл и осуществляет переход только при CX=0. Ее удобно использовать в начале цикла, особенно в ситуации, когда цикл может не выполняться ни разу. Если команда LOOP выполняет постпроверку содержимого CX, то команда JCXZ выполняет его предпроверку.

Фактически команда LOOP label эквивалентна двум командам

DEC CX

JNZ label

и ее использование кроме удобства программирования позволяет экономить один байт объектного кода и один такт T. При большом числе повторений цикла экономия времени может оказаться значительной.

Команда LOOPE/LOOPZ по сравнению с командой LOOP вводит дополнительное условие повторения цикла: ZF=1. Это позволяет выходить из вычислительного цикла как по окончании заданного числа циклов, так и при получении ненулевого результата (или неравенстве сравниваемых операндов). Команда LOOPNE/LOOPNZ вводит дополнительное условие противоположного смысла: ZF=0, которое позволяет прекращать цикл при получении нулевого результата (или по равенству сравниваемых операндов).

Команды обработки цепочек. Имеется пять однобайтовых команд (рис. 2.16), предназначенных для обработки одного элемента цепочки. Напомним, что под цепочкой (строкой) понимается последовательность любых контекстно связанных байтов или слов, находящихся в смежных ячейках памяти. Цепочечной команде может предшествовать однобайтовый префикс повторения REP, который вызывает повторение действия команды над следующим элементом. Это позволяет обрабатывать цепочки значительно быстрее, чем при организации программного цикла. Повторение рассчитано на максимальную длину цепочки 64 Кбайт и может заканчиваться по одному или двум условиям. Требуемое число повторений указывается в регистре CX, содержимое которого автоматически декрементируется при каждом повторении операции. Когда содержимое CX становится равным нулю, МП выполняет следующую команду. При выполнении повторяющейся цепочечной команды МП воспринимает внешние прерывания.

В качестве операндов команды могут иметь цепочку-источник, цепочку-получатель или то и другое одновременно. Цепочка-источник по умолчанию размещается в текущем сегменте данных, и все элементы адресуются регистром SI. Имеется возможность использовать другой сегмент, указав его в префиксе смены сегмента. Цепочка-получатель всегда размещается в дополнительном сегменте, и ее элементы адресуются регистром DI. Для начальной загрузки адресов в регистры можно использовать команды LEA, LDS и LES.

При выполнении цепочечной команды содержимое SI и DI автоматически изменяется на ± 1 (при обработке байта) или на ± 2 (при обработке слова), чтобы адресовать следующие элементы цепочек. Флаг DF определяет направление изменения адресов: автоинкремент (DF=0) или автодекремент (DF=1).

Каждая цепочечная команда способна оперировать как с байтами, так и со словами, причем тип элементов цепочек ассемблер может определить по атрибутам операндов. Однако многие версии ассемблера допускают указание типа элементов путем добавления букв B (байт) или W (слово) к mnemonic обозначениям цепочечных команд: MOSB или MOVSW, CMPSB или CMPSW и т. д. При этом операнды в команде становятся необязательными (ассемблер их игнорирует, так как адресация цепочек известна заранее) и могут указываться только для удобства программирования.

Префикс повторения (см. рис.2.16) обеспечивает повторяющееся (циклическое) выполнение цепочечной команды. При отсутствии префикса команда оперирует только одним элементом цепочки. Префикс повторения имеет пять мнемоник: REP/REPE/REPZ и REPNE/REPNZ, которые определяют только два объектных кода, различающихся однобитным полем Z, и введены для лучшего отражения содержательного смысла команды. Префикс повторения не влияет на флаги. Он обеспечивает автоматическое декрементирование регистра CX на каждом повторении команды и проверку содержимого CX на ноль.

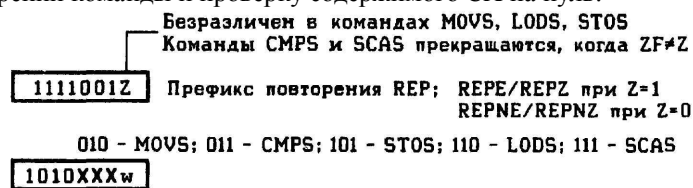


Рис. 2.16. Команды обработки цепочек

Префикс REP используется с командами пересылки MOVS и STOS и инициирует действие “повторять, пока не будет достигнут конец цепочки”, т. е. пока CX≠0. Префиксы REPE и REPZ действуют аналогично и имеют такой же машинный код, как и префикс REP. Они используются с командами сравнения CMPS и SCAS и оперируют с флагом ZF, состояние которого определяется результатом исполнения этих команд, причем появление ZF=1 останавливает процесс повторения команд. Префиксы REPNE и REPNZ также применяются командами CMPS и SCAS, но используют противоположное условие ZF=0. Наличие дополнительных проверяемых условий позволяет окончить выполнение этих команд по результату вычислений до завершения перебора всех элементов цепочки.

Таблица 2.12

Команда	Время выполнения	
	без префикса повторения	с префиксом повторения
MOVS	18	9 + 17n
CMPS	18	9 + 17n
SCAS	15	9 + 15n
LODS	12	9 + 13n
STOS	11	9 + 10n

Цепочечные команды пересылки MOVS, LODS и STOS на флаги не влияют, а команды неразрушающего сравнения CMPS и SCAS устанавливают значения флагов в соответствии с разностью операндов, сохраняя значения самих операндов. Время выполнения цепочечных команд (в тактах синхронизации) указано в табл. 2.12, где n - число повторений, которое предварительно заносится в регистр CX.

Команды с префиксом повторения при обработке длинных цепочек могут выполняться в течение значительного времени. Поэтому для сокращения времени реакции на внешние прерывания МП воспринимает внешние сигналы прерывания после обработки каждого элемента цепочки. (Сигнал INTR=1

должен установиться не более чем за один такт до окончания цикла обработки).

После возврата из прерывания операция возобновляется, как обычно, с точки прерывания. Однако во время обработки прерывания МП помнит действие только одного префикса, непосредственно предшествующего команде. Поэтому, если с цепочечной командой используется несколько префиксов (префикс замены сегмента, префикс блокировки), необходимо запрещать прерывания на время ее выполнения.

Обобщенные обозначения цепочечных команд и выполняемые ими действия при различных префиксах приведены в табл. 2.13.

Таблица 2.13.

Команда, её действие	Префикс повторения	Описание действия
MOVS dst ← src	-	Пересылка из цепочки src, адресуемой регистром SI (сегмент данных) в цепочку dst, адресуемую регистром DI (дополнительный сегмент)
CMPS dst,src; src - dst	REP	Блочная пересылка память-память
	REPE/REPZ	Вычитание элемента цепочки dst из элемента цепочки src (неразрушающее сравнение)
	REPNE/REPZ	Сравнение элементов до обнаружения одинаковых элементов или до окончания цепочек
SCAS dat; ac - dst	-	Сравнение элементов до обнаружения различающихся элементов или до окончания цепочек
	REPE/REPZ	Вычитание элемента цепочки dst, адресуемой регистром DI, из содержимого AL или AX (неразрушающее сравнение)
	REPNE/REPZ	Поиск элемента цепочки со значением, отличающимся от ac
LODS src; ac ← src	-	Поиск элемента цепочки со значением, совпадающим с ac
	-	Загрузка аккумулятора элементом цепочки, адресуемым регистром SI (префикс повторения обычно не используется)
STOS dst; dst ← ac	-	Запоминание содержимого аккумулятора AL или AX в элементе цепочки, адресуемом регистром DI
	REP	Инициализация цепочки на фиксированное значение, содержащееся в аккумуляторе

Команду CMPS удобно применять для нахождения одинаковых (REPE CMPS) или различающихся (REPNE CMPS) элементов цепочек. Достижение конца цепочки при выполнении этих команд соответственно означает, что цепочки полностью различны (не содержат одинаковых элементов) или полностью совпадают (не содержат различающихся элементов). Команда SCAS производит сравнение элемента цепочки с некоторым эталоном, содержащимся в аккумуляторе. Вариант REPE SCAS осуществляет просмотр цепочки до тех пор, пока значение элемента не отличается от эталона или не достигается конец цепочки, а вариант REPNE SCAS — просмотр цепочки до обнаружения элемента, равного эталону, или до достижения конца цепочки.

Команду LODS удобно использовать в программных циклах вместо двух команд: MOV, ax,src и INC SI (или DEC SI в зависимости от направления продвижения по цепочке). Отметим, что команда LODS оперирует элементами цепочки, расположенной в сегменте данных, причем имеется возможность замены сегмента, а команда STOS оперирует элементами цепочки, которая всегда находится в дополнительном сегменте. В последнем случае префикс замены сегмента не указывается, а при его наличии — игнорируется.

Дополнительная литература

1. Брябин В.М. Программное обеспечение персональных ЭВМ. - М.: Наука, 1988.
2. Григорьев В.Л. Программирование однокристальных микропроцессоров.- М.: Энергоатомиздат, 1987.
3. Дао Л. Программирование микропроцессора 8088/Пер. с англ.- М.: Мир,1988.
4. Казаринов Ю. М., Номоконов В. Н., Филиппов Ф. В. Применение микропроцессоров и микроЭВМ в радиотехнических системах.- М.: Высшая школа, 1988.
5. Лю Ю-Чжен, Гибсон Г. Микропроцессоры семейства 8086/8088. Архитектура, программирование и проектирование микрокомпьютерных систем /Пер. с англ.- М.: Радио и связь, 1987.

9.3.Список литературы, используемой при подготовке к лабораторным работам.

- 1) Электронный учебник PC BOOK (Текстовый файл формата DOS, 1.2 Мбайт).
- 2) Лекции по курсу «Адаптеры и контроллеры».
- 3) Микропроцессоры и микроЭВМ в системах автоматического управления. Справочник под редакцией С.Т. Хвоща. Машиностроение, 1987г.
- 4) Аппаратура персональных компьютеров и ее программирование IBM/PC/XT/AT и PS/2. М: Радио и связь, 1995 г.
- 5) Финогенов К.Г.. Самоучитель по системным функциям MS DOS. М: Радио и связь, 1995 г.
- 6). Гук М. Аппаратные средства PC. Энциклопедия - СПб. Питер Ком, 1998.