

Ю.С.Лукач, А.Е.Сибиряков

АРХИТЕКТУРА ВВОДА - ВЫВОДА
ПЕРСОНАЛЬНЫХ ЭВМ IBM PC

(С) Инженерно-техническое бюро

Свердловск
1990

Содержание

Введение	
Обозначения	
1. Таблица портов ввода-вывода	
2. Контроллер прямого доступа к памяти	
2.1. Принципы работы контроллера ПДП	
2.2. Типы передач	
2.3. Описание внутренних регистров ПДП	
2.4. Программное управление контроллером ПДП	
2.4.1. Организация и управление ПДП на XT-подобных ПЭВМ	
2.4.2. Организация и управление ПДП на AT-подобных ПЭВМ	
3. Программируемый контроллер прерываний	
3.1. Описание основных элементов ПКП	
3.2. Режимы работы ПКП	
3.3. Программирование ПКП.	
4. Таймер и генерация звука	
4.1. Программируемый таймер 8253	
4.2. Таймер на системной плате IBM PC	
4.3. Генерация звука	
5. Программируемый периферийный интерфейс	
5.1. Описание ППИ 8255	
5.2. Использование ППИ в IBM PC	
6. Клавиатура	
6.1. Клавиатура IBM PC/XT	
6.2. Клавиатура IBM PC AT	
7. Дисплеи	
7.1. Общее описание	
7.2. Контроллеры ЭЛТ	
7.3. Атрибуты символов	
7.4. Монохромный графический адаптер MGA	
7.5. Цветной графический адаптер CGA	
7.6. Расширенный графический адаптер EGA	
7.7. Графический видеомассив VGA	
8. Контроллер накопителей на гибких дисках	
8.1. Основные сведения о контроллере	
8.2. Порты контроллера накопителей на гибких дисках.	
8.3. Команды управления накопителями	
8.3.1. Перечень команд контроллера НГМД	
8.3.2. Структура байтов состояния	
8.4. Формат дорожки.	
8.5. Отличия контроллера для AT-подобных ПЭВМ.	
9. Контроллер накопителей на жестких дисках	
9.1. Контроллер жестких дисков для XT-подобных ПЭВМ.	

9.1.1.	Порты ввода-вывода	
9.1.2.	Общий формат команды и терминология	
9.1.3.	Описание команд	
9.1.4.	Разметка диска	
9.2.	Контроллер жестких дисков для AT-подобных ПЭВМ.	
9.2.1.	Описание портов ввода-вывода	
9.2.2.	Система команд	
10.	Печатающее устройство	
11.	Последовательный интерфейс RS-232	
11.1.	Общие сведения	
11.2.	Описание портов	
11.3.	Порядок инициализации 8250	
12.	Игровой адаптер	
13.	CMOS	
13.1.	Доступ к CMOS	
13.2.	Содержимое CMOS	
14.	Прочие порты	
Приложение 1.	Скан-коды клавиш	
Приложение 2.	Типы жестких дисков	
Приложение 3.	Подготовка ПЭВМ к работе	
Глоссарий	

Введение

Данный документ содержит техническое описание устройств ввода-вывода, входящих в состав ПЭВМ семейства IBM PC, включая ПЭВМ IBM PC XT, AT и PS/2. Он основан на технической документации фирм - производителей периферийной аппаратуры, охватывающей несколько десятков руководств. В частности, описание микросхем, размещенных на системной плате, базируется на справочниках фирмы INTEL (Intel Handbooks).

Дополнительными источниками информации послужили многочисленные монографии, журнальные статьи и диалоговые справочные системы (такие, как TECH Help!, Norton Guide и др.). Кроме того, данное описание учитывает опыт авторов по разработке и сопровождению программ в среде MS-DOS. Этот опыт позволил, во-первых, исправить отдельные неточности и недоговоренности в использованных материалах, и, во-вторых, описать ряд недокументированных особенностей оборудования IBM PC.

Вся приведенная информация подвергалась максимальной проверке на ЭВМ; однако, авторы предполагают, что отдельные нюансы описаний, заимствованных из фирменной документации, могут содержать ошибки. Такие "подозрительные" моменты мы, по мере возможности, старались явно оговаривать в соответствующих местах справочника.

Для понимания изложенного здесь материала требуется владение основами программирования на языке ассемблера для IBM PC (знание архитектуры процессора и директив ассемблера).

Авторы будут благодарны за любые указания на неточности, замечания и дополнения, которые мы просим присылать по адресу:

620075, г. Свердловск,
ул. Луначарского 81,
Инженерно-техническое бюро.

тел. 56-92-54, 56-92-41.

Обозначения

PC, XT, AT, PCjr, PC Convertible и PS/2 - это имена компьютеров семейства IBM PC. Все они более или менее совместимы между собой на уровне прикладных пакетов, но имеют определенные аппаратные различия:

PC: центральный процессор 8088, тактовая частота 4.77 МГц, 5 позиций для адаптерных плат.

PCjr: ЦП 8088, нет места для плат. Выпущена мизерным тиражом и не совместима с остальными IBM PC. В нашей серии не описывается.

XT: ЦП 8088, жесткий диск, 8 позиций для адаптерных плат. В остальном аналогична PC.

Turbo PC: ПЭВМ других фирм (не IBM), совместимые с PC и XT, но имеющий ЦП 8086 или V30 фирмы NEC, тактовая частота которых 6 или 8 МГц. Обычно имеет 8 позиций для плат.

XT-286: ЦП 80286, 8 позиций для плат, частота 6 МГц без холостых циклов (скорость как у AT с 8 МГц). Платы от AT к ней не подходят, от PC подходят, но не позволяют использовать преимущества 16-разрядной системной шины.

AT: ЦП 80286, 8 позиций для плат, тактовая частота 6 или 8 МГц.

PC Convertible: Портативный вариант AT. Отличия: плазменный (LCD) дисплей, 3.5-дюймовые дисководы, нет позиций для плат расширения.

PS/2: Новое семейство ПЭВМ фирмы IBM. Модели 25 и 30 - это улучшенные версии старых PC, совместимые с ними по адаптерным платам. Модели 50 и 60 - ЦП 80286, 32-разрядная шина MicroChannel, совместимости со старыми платами нет. Модель 80 имеет ЦП 80386. Все модели оснащены дисководами размером 3.5 дюйма.

AT-386: ПЭВМ третьих фирм, совместимая с AT, но имеющая ЦП 80386.

Если рассматриваемые устройства зависят от типа ПЭВМ, то используются следующие пометки:

PC/XT	относится к ПЭВМ IBM PC и XT, включая Portable PC
AT	относится к ПЭВМ IBM PC AT
PS/2	относится к семейству PS/2

Отсутствие таких указаний означает, что данный материал применим ко всем ПЭВМ фирмы IBM.

Десятичные числа записываются обычным образом, шестнадцатичные отличаются добавлением буквы "h" в конце. Например, 256 и 100h означают десятичное число 256.

Все размеры полей в управляющих структурах приведены в байтах, если не оговорено противное.

В некоторых случаях алгоритмы сопровождаются формулами. Операции в формулах обозначаются, как принято в языке Си, а именно:

арифметические: +, -, *, /, % (взятие остатка)
 битовые: & (и), | (или), ~ (не)
 сдвиги: << и >>.

1. Таблица портов ввода-вывода

Приведенная ниже таблица содержит распределение адресного пространства портов ввода-вывода для IBM PC/XT, AT и PS/2. При этом:

- порты 0 - FFh отведены для системной платы;
- порты 100h - 3FFh отведены для контроллеров устройств;
- порты, начиная с 400h, недоступны для системной шины.

AT/PS-2	PC/XT	Описание	См. раздел
000-01F	000-00F	Контроллер ПДП N 1, 8237A-5	2
020-03F	020-021	Контроллер прерываний N 1, 8259A	3
040-05F	040-043	Таймер (PC/XT:8253-5, AT:8254-2)	4
	060-063	Программируемый интерфейс периферии 8255	5
060-06F		Контроллер клавиатуры AT 8042	6
070-07F		Память CMOS и маска NMI	13
080		Диагностический регистр	14
080-08F	080-083	Регистры страниц ПДП 74LS612	2
090-097		Блок управления каналами PS/2	—
	0A0	Маска NMI	14
0A0-0BF		Контроллер прерываний N 2, 8259A	3
0C0-0DF		Контроллер ПДП N 2, 8237A-5	2
0F0-0FF		Сопроцессор 80287	14
101-107		Выбор программируемых опций PS/2	—
170-177		Жесткий диск N 2	9
1F0-1F7		Жесткий диск N 1	9
200-207	200-20F	Игровой адаптер	12
	210-217	Блок расширений	14
238-23F		Стык RS-232 N 4 (в PS/2)	11
278-27F	278-27F	Параллельный принтер N2 (N 3 при MGA)	10
2C0-2DF	2C0-2DF	Расширенный графический адаптер (EGA) N 2	7
2F8-2FF	2F8-2FF	Стык RS-232 N 2	11
300-31F		Плата прототипа	—
	320-32F	Жесткий диск XT	9
338-33F		Стык RS-232 N 3 (в RS/2)	11
370-377		Контроллер НГМД N 2	8
378-37F	378-37F	Параллельный принтер N 1 (N2 при MGA)	10
380-38F	380-38F	Синхронные адаптеры: SDLC или BSC N 2	—
3A0-3AF	3A0-3A9	Синхронный адаптер BSC N 1	—
3B0-3BF	3B0-3BF	Монохромный адаптер (MGA) + принтер N 1	7,10
3C0-3CF	3C0-3CF	Расширенный графический адаптер (EGA) N 1	7
3D0-3DF	3D0-3DF	Цветной графический адаптер (CGA) и EGA	7
3F0-3F7	3F0-3F7	Контроллер НГМД N 1	8
3F8-3FF	3F8-3FF	Стык RS-232 N 1	11

Примечание: В таблицу не включены устройства, базовый адрес ввода-вывода которых может настраиваться (сетевые адаптеры, контроллеры стримерных лент и т.д.).

2. Контроллер прямого доступа к памяти

Контроллер прямого доступа к памяти (ПДП, DMA - Direct Memory Access) обеспечивает высокоскоростной обмен данными между устройствами ввода-вывода и ОЗУ без использования центрального процессора, что позволяет освободить процессор для выполнения вычислений параллельно с обменом и независимо от него. Наиболее часто возможности ПДП используются при работе с дисковыми накопителями, однако реализовано использование ПДП адаптерами накопителей на магнитной ленте и рядом других устройств. Ощутимые преимущества дает использование ПДП в процессе обмена с устройствами, принимающими или передающими данные достаточно большими порциями с высокой скоростью.

В IBM PC-подобных компьютерах функции контроллера ПДП выполняет микросхема 8237A фирмы INTEL (советский аналог КР580ВТ57) или ее аналоги 8237A-4 и 8237A-5, работающие с тактовой частотой 4 и 5 МГц соответственно (стандартная микросхема 8237A работает на частоте 3 МГц). Контроллер имеет 4 независимых канала, каждый из которых может обслуживать одно периферийное устройство.

2.1. Принципы работы контроллера ПДП

В работе ПДП различаются 2 главных цикла: цикл ожидания (Idle cycle) и активный цикл (Active cycle). Каждый цикл подразделяется на ряд состояний, занимающих по времени один период часов (тик). Из цикла ожидания контроллер может быть переведен в состояние программирования (Program Condition) путем подачи на вход RESET сигнала высокого уровня, длительностью не менее 300 нс и следующей за ним подачи сигнала низкого уровня (уровня 0) на вывод CS (Chip Select). В состоянии программирования контроллер будет находиться до тех пор, пока на выводе CS сохранится сигнал низкого уровня. В процессе программирования контроллеру задаются:

- начальный адрес памяти для обмена;
- уменьшенное на единицу число передаваемых байтов;
- направление обмена,

а также устанавливаются требуемые режимы работы (разрешить или запретить циклическое изменение приоритетов, автоинициализацию, задать направление изменения адреса при обмене и т. д.).

Загрузка 16-разрядных регистров контроллера осуществляется через 8-разрядные порты ввода-вывода. Перед загрузкой первого (младшего) байта должен быть сброшен (очищен) триггер-зашелка (триггер первый/последний, First/Last flip-flop), который изменяет свое состояние после вывода в порт первого байта и таким образом дает возможность следующей командой вывода в тот же порт загрузить старший байт соответствующего регистра.

Запрограммированный канал должен быть демаскирован (бит маски канала устанавливается при этом в 0), после чего он может принимать сигналы "Запрос на ПДП", генерируемые тем внешним устройством, которое обслуживается через этот канал. Сигнал "Запрос на ПДП" может быть также инициирован установкой в 1 бита запроса данного канала в регистре запросов контроллера. После появления сигнала запроса контроллер входит в активный цикл, в котором выполняется обмен данными. Обмен может осуществляться в одном из четырех режимов:

1. Режим одиночной передачи (Signle Transfer Mode). После каждого цикла передачи контроллер освобождает шину процессору,

но сразу же начинает проверку сигналов запроса и, как только обнаруживает активный сигнал запроса, инициирует следующий цикл передачи.

2. Режим блочной передачи (Block Transfer Mode). В этом режиме наличие сигнала запроса требуется только до момента выдачи контроллером сигнала "Подтверждение запроса на ПДП" (DACK), после чего шина не освобождается вплоть до завершения передачи всего блока.

3. Режим передачи по требованию (Demand Transfer Mode). Данный режим является промежуточным между двумя первыми: передача идет непрерывно до тех пор, пока активен сигнал запроса, состояние которого проверяется после каждого цикла передачи. Как только устройство не может продолжить передачу, сигнал запроса сбрасывается им и контроллер приостанавливает работу. Этот режим применяется для обмена с медленными устройствами, не позволяющими по своим временным характеристикам работать с ПДП в режиме блочной передачи.

4. Каскадный режим (Cascade Mode). Режим позволяет включить в подсистему ПДП более одного контроллера в тех случаях, когда недостаточно четырех каналов ПДП. В этом режиме один из каналов ведущего контроллера используется для каскадирования с контроллером второго уровня. Для работы в каскаде сигнал HRQ ("Запрос на захват") ведомого контроллера подается на вход DREG ("Запрос на канал ПДП") ведущего, а сигнал DACK ("Подтверждение запроса") ведущего подается на вход HDLA ("Подтверждение захвата") ведомого.

Такая схема подключения аналогична подключению ведущего (первого) контроллера к микропроцессору, с которым он обменивается сигналами HRQ и HDLA.

2.2. Типы передач

1. Передача память-память (Memory-to-memory DMA).

Используется для передачи блока данных из одного места памяти в другое. Исходный адрес определяется в регистрах нулевого канала, выходной - в регистрах первого канала. Число циклов обмена (число байт минус 1) задается в регистре числа циклов канала 1. Передача происходит с использованием рабочего регистра контроллера в качестве промежуточного звена для хранения информации. При передаче память-память может быть задан специальный режим фиксации адреса (Address hold), при котором значение текущего адреса в регистре нулевого канала не изменяется, при этом весь выходной блок памяти заполняется одним и тем же элементом данных, находящимся по заданному адресу.

2. Автоинициализация (автозагрузка, Autoinitialization).

После завершения обычной передачи использованный канал ПДП маскируется и должен быть перепрограммирован для дальнейшей работы с ним. При автоинициализации маскировка канала после окончания передачи не происходит, а регистры текущего адреса и счетчик циклов автоматически загружаются из соответствующих регистров с начальными значениями. Таким образом для продолжения (повторения) обмена достаточно выставить сигнал запроса на ПДП по данному каналу.

3. Режим фиксированных приоритетов.

В этом режиме канал 0 всегда имеет максимальный приоритет, а канал 3 - минимальный. Это означает, что любая передача по каналу с более высоким приоритетом будет выполняться раньше, чем

по каналу с более низким приоритетом.

4. Циклический сдвиг приоритетов.

Позволяет избежать "забивания" шины одним каналом при одновременной передаче по нескольким каналам. Каждому каналу, по которому прошла передача, автоматически присваивается низший приоритет, после чего право на передачу получает канал с наивысшим приоритетом, для которого передача в данный момент возможна. Таким образом, если в начале работы распределение приоритетов было обычным (канал 0 - наивысший), и пришли сигналы запроса на ПДП по 1-му и 2-му каналам, то сначала будет выполняться передача по первому каналу, затем он получит низший приоритет (а канал 2, соответственно, высший, т. к. сдвиг приоритетов циклический) и передача выполнится по 2-му каналу, который затем получит низший приоритет, а высший приоритет получит, соответственно, канал 3, который и будет обладать преимущественным правом на передачу.

5. Сжатие времени передачи (Compressed transfer timing).

В случае, если временные характеристики быстродействия обменивающихся устройств совпадают, ПДП может сократить время выполнения каждого такта передачи на 2 цикла часов за счет тактов ожидания, входящих в каждый цикл передачи.

2.3. Описание внутренних регистров ПДП

Контроллер имеет 344 бита внутренней памяти, организованной в виде регистров. Описание внутренних регистров ПДП приведено в таблице.

Наименование регистра	Разрядность (бит)	Число регистров
Регистр начального адреса (Base Address Register)	16	4
Регистр начального счетчика циклов (Base Word Count Register)	16	4
Регистр текущего адреса (Current Address Register)	16	4
Регистр текущего счетчика циклов (Current Word Count Register)	16	4
Рабочий регистр адреса (Temporary Address Register)	16	1
Рабочий регистр счетчика циклов (Temporary Word Count Register)	16	1
Регистр состояния (Status Register)	8	1
Регистр команд (Command Register)	8	1
Регистр режима (Mode Register)	6	4
Рабочий регистр (Temporary Register)	8	1

Регистр масок (Mask Register)	4	1
Регистр запросов (Request Register)	4	1

Регистр начального адреса (Base Address Register).

В этом регистре задается стартовый адрес ОЗУ, с которого начинается передача. Регистр содержит 16 разрядов и определяет адрес внутри заданной страницы памяти размером 64К. Задание номера страницы памяти осуществляется через специальные страничные регистры (Page Registers), поддерживаемые внешней логикой. Каждый канал ПДП имеет свой регистр начального адреса и страничный регистр. Такое деление памяти на страницы не позволяет осуществить обмен с блоком памяти, находящимся на пересечении двух страниц. Каждая страница начинается с сегментного адреса, кратного 1000h (0, 1000h, 2000h, ..., 9000h).

Регистр начального счетчика циклов (Base Word Count Register).

В этом регистре задается начальное число циклов передачи для программируемого канала. Фактическое число передаваемых во время работы ПДП элементов данных на единицу превышает заданное число циклов, т. е. если Вы задаете 100 циклов передачи, а размер элемента будет равен 1 байту, то за сеанс обмена будет передан 101 байт информации.

Регистр текущего адреса (Current Address Register).

Начальное значение заносится в этот регистр одновременно с регистром начального адреса. В дальнейшем в ходе передачи значение текущего адреса автоматически увеличивается или уменьшается (конкретное направление изменения задается при программировании в регистре режима). Если разрешена автоинициализация, то после окончания передачи в регистр автоматически устанавливается значение из регистра начального адреса.

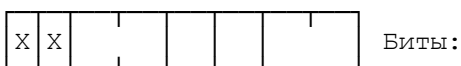
Регистр текущего счетчика циклов (Current Word Count Register).

Регистр содержит текущее значение счетчика циклов (число оставшихся циклов передачи). Отображаемое в нем число циклов всегда на единицу меньше числа еще не переданных элементов данных, так как изменение значения в этом регистре производится в конце цикла передачи, уже после фактической передачи элемента данных, а конец передачи фиксируется в момент переполнения счетчика (изменение его значения с 0 на 0FFFFh).

Регистр режима (Mode Register).

Данный регистр задает режимы работы своего канала контроллера.

7 6 5 4 3 2 1 0



- > 0-1: Режим работы: 00=проверка, 01=запись (в память), 10=чтение (из памяти), 11=недопустимая комбинация
- > 2: Автоинициализация: 1=разрешена, 0=запрещена
- > 3: Изменение текущего адреса при обмене: 0=увеличение, 1=уменьшение
- > 4-5: Тип передачи:

00=Режим передачи по требованию,
 01=Режим одиночной передачи,
 10=Режим блочной передачи,
 11=Каскадный режим.

Каждый из четырех каналов ПДП имеет свой набор регистров, описанных выше. Кроме того, имеется следующий набор регистров, общих для всех каналов.

Регистр команд (Command Register).

Этот 8-битный регистр управляет работой контроллера. Он программируется, когда контроллер находится в состоянии программирования и очищается командами сброса "Reset" и "Master Clear". Назначение битов регистра команд приведено на рисунке.



Регистр состояния (Status Register).

Регистр отражает текущее состояние запросов и передач по всем четырем каналам. Биты 0 - 3 устанавливаются в единицу после завершения передачи по каналам 0 - 3 (бит 0 - канал 0, бит 1 - канал 1 и т.д.), если не задан режим автоинициализации. Эти биты очищаются после команды сброса контроллера и после каждой операции считывания состояния из регистра состояния. Биты 4 - 7 указывают по какому из каналов 0 - 3 активен в текущий момент сигнал запроса на ПДП.

Регистр масок (Mask Register).

Каждый бит этого 4-битового регистра маскирует/демаскирует свой канал ПДП, при этом значение 1 маскирует канал, значение 0 демаскирует канал и разрешает прием сигнала запроса по этому каналу.

Регистр запросов (Request Register).

Сигнал запроса на ПДП (DREQ) может быть издан как обслуживаемым устройством, так и программно. Для программного издания сигнала запроса по одному из 4-х каналов ПДП необходимо установить соответствующий бит в 4-разрядном регистре запросов. Запрос на ПДП может быть отменен записью нулевого значения в соответствующий бит регистра. Бит запроса очищается автоматически при окончании передачи по данному каналу. Все биты

запросов очищаются при сбросе контроллера. Для того, чтобы воспринимать программные запросы на ПДП, канал должен находиться в режиме блоковой передачи.

Рабочий регистр (Temporary Register).

Этот 8-разрядный регистр используется для хранения элемента данных, передаваемого в режиме фиксированного адреса при передаче память-память или для временного хранения передаваемого байта при всех остальных режимах передачи.

2.4. Программное управление контроллером ПДП

Программное управление контроллером ПДП осуществляется через порты ввода-вывода. Доступ к каждому регистру контроллера может быть осуществлен через свои порты ввода-вывода. В разделах 2.4.1 и 2.4.2 описывается программное управление контроллерами на XT- и AT-подобных ПЭВМ соответственно.

2.4.1. Организация и управление ПДП на XT-подобных ПЭВМ

IBM PC XT и XT-совместимые ПЭВМ имеют один 4-канальный контроллер ПДП 8237A. Назначение каналов следующее:

- 0 - обмен память-память;
- 1 - свободен (может использоваться сетевыми адаптерами);
- 2 - обмен с контроллером гибких дисков;
- 3 - обмен с контроллером жестких дисков.

В таблице приведено описание портов ввода-вывода, предназначенных для управления контроллером ПДП.

Порт	Режим	Назначение
0h	запись	Запись начального адреса в регистр начального адреса и регистр текущего адреса канала 0
2h	запись	то же для канала 1
4h	запись	то же для канала 2
6h	запись	то же для канала 3
0h	чтение	Чтение начального адреса из регистра начального адреса канала 0
2h	чтение	то же для канала 1
4h	чтение	то же для канала 2
6h	чтение	то же для канала 3
1h	запись	Запись в регистр начального счетчика циклов и в регистр текущего счетчика циклов канала 0
3h	запись	то же для канала 1
5h	запись	то же для канала 2
7h	запись	то же для канала 3

Порт	Режим	Назначение
1h	чтение	Чтение текущего значения из регистра текущего счетчика циклов канала 0
3h	чтение	то же для канала 1
5h	чтение	то же для канала 2
7h	чтение	то же для канала 3
8h	запись	Запись регистра команд ПДП
8h	чтение	Чтение регистра состояния ПДП
9h	запись	Запись в регистр запросов ПДП
0Ah	запись	Запись бита маски для одного из каналов ПДП
0Bh	запись	Запись регистра режимов для одного из каналов ПДП
0Ch	запись	Очистка (сброс) триггера-защелки (триггера первого/последнего)
0Dh	запись	Программный сброс контроллера
0Eh	запись	Очистка битов масок всех 4-х каналов
0Fh	запись	Запись регистра масок для всех 4-х каналов
0Dh или 0Fh	чтение	Чтение рабочего регистра ПДП
81h	запись	Задание номера страницы для канала 2
82h	запись	то же для канала 3
83h	запись	то же для канала 1
87h	запись	то же для канала 0

Порты 0h - 7h предназначены для записи исходных значений в регистры начального и текущего адреса, начального и текущего счетчика циклов для всех 4-х каналов. Так как порты восьмиразрядные, а регистры, в которые через них заносятся данные, 16-разрядные, то запись производится в два приема. Перед первой командой вывода в требуемый порт необходимо сбросить триггер-защелку, для чего выполняется команда вывода произвольного значения в порт 0Ch, после чего в требуемый порт выводится младший байт 16-разрядного значения и затем старший байт следующей команды вывода в тот же порт.

Вывод в порт 8h позволяет занести значение в регистр команд ПДП. Описание битов регистра команд приведено в п.2.3.

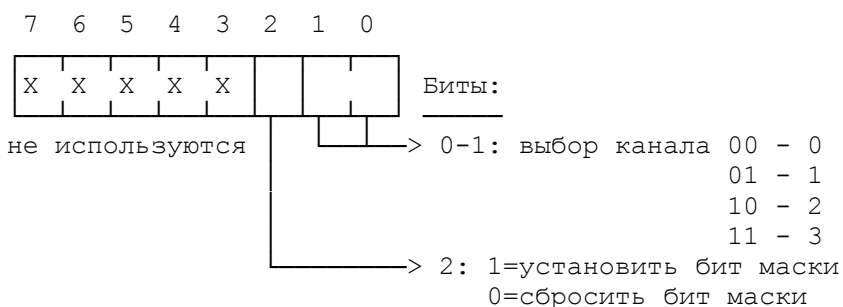
Чтение из порта 8h считывает регистр состояния ПДП. Описание битов регистра состояния приведено в п.2.3.

Запись в порт 9h позволяет установить или сбросить бит запроса в регистр запросов для одного из каналов. Формат команды

следующий:



Запись в порт 0Ah позволяет установить или сбросить бит маски в регистре масок для одного из каналов. Формат команды:



Запись в порт 0Bh устанавливает значение в регистре режимов одного из 4-х каналов ПДП. Биты 0 и 1 задают номер канала (00 - 0, 01 - 1, 10 - 2, 11 - 3). В биты 2 - 7 заносятся значения, передаваемые соответственно в биты 0 - 6 регистра режимов. Описание битов регистра режимов приведено в п. 2.3.

Запись в порт 0Dh задает программный сброс контроллера (Master Clear). Вывод любого байта в этот порт имеет тот же эффект, что и аппаратный сброс контроллера. При программном сбросе очищаются регистры команд, состояния, запросов и рабочий регистр. Так же сбрасывается триггер-защелка и устанавливается все биты масок в регистре масок. После программного сброса контроллер переходит в цикл ожидания.

Вывод любого байта в порт 0Eh очищает регистр масок - сбрасывает биты масок всех 4-х каналов ПДП и таким образом разрешает прием запросов на ПДП по всем каналам.

Через порт 0Fh можно задать произвольное значение регистра масок ПДП. Для этого необходимо в битах 0 - 3 регистра AL установить требуемое значение масок каналов 0 - 3 соответственно и вывести это значение в порт.

Страничные регистры ПДП предназначены для задания параметра страницы памяти, с которой будет производиться обмен. Под номером страницы понимаются старшие 4 бита полного 20-битового адреса ОЗУ, т.е. начальные сегментные адреса страниц будут кратны 1000h (0, 1000h, 2000h, ..., 9000h). Напомним, что архитектура ПДП позволяет работать только со страницей памяти размером 64К, а логика, обеспечивающая переключение страниц устроена так, что страницы имеют жесткие границы, описанные выше. Из-за этой особенности невозможно с помощью ПДП осуществить обмен с блоками памяти, пересекающими границу между двумя такими страницами.

2.4.2. Организация и управление ПДП на АТ-подобных ПЭВМ

IBM PC AT и AT-подобные ПЭВМ имеют два контроллера ПДП 8237A, работающие в каскадном режиме.

Назначение каналов следующие:

- 0 - свободен;
- 1 - адаптер SDLC;
- 2 - контроллер накопителей на гибких дисках;
- 3 - контроллер накопителей на жестких дисках;
- 4 - каскад с первым контроллером ПДП;
- 5, 6, 7 - свободны.

Управление каналами 0 - 3 осуществляется аналогично PC/XT. Каналы 4 - 7 предназначены для обмена 16-разрядными словами. В связи с этим возникает ряд отличий в работе с этими каналами:

- бит 0 в данных, заносимых в регистры начального и текущего адреса, всегда подразумевается равным 0, поэтому через эти регистры передаются биты 1 - 16 полного 23-разрядного адреса (а не биты 0 - 15 полного 20-разрядного адреса, как это реализовано на XT-подобных ПЭВМ), по этой же причине в страничные регистры каналов 4 - 7 заносятся биты 17 - 23 полного адреса, а не биты 16 - 23, как это надо сделать при работе с каналами 0 - 3;

- поскольку передача осуществляется 16-разрядными словами, в регистры текущего и начального счетчика циклов заносится не число байт, а число слов, уменьшенное на единицу;

- размеры страниц памяти, в пределах которых возможен обмен в течение одной передачи составляют 2000h байтов.

В таблице приведено описание портов ввода-вывода, предназначенных для управления вторым контроллером ПДП.

Порт	Режим	Назначение
0C0h	запись	Запись начального адреса в регистр начального адреса и регистр текущего адреса канала 4
0C4h	запись	то же для канала 5
0C8h	запись	то же для канала 6
0CCh	запись	то же для канала 7
0C0h	чтение	Чтение начального адреса из регистра начального адреса канала 4
0C4h	чтение	то же для канала 5
0C8h	чтение	то же для канала 6
0CCh	чтение	то же для канала 7
0C2h	запись	Запись в регистр начального счетчика циклов и в регистр текущего счетчика циклов канала 4
0C6h	запись	то же для канала 5
0CAh	запись	то же для канала 6
0CEh	запись	то же для канала 7

0C2h	чтение	Чтение текущего значения из регистра текущего счетчика циклов канала 4
0C6h	чтение	то же для канала 5
0Cah	чтение	то же для канала 6
0CCh	чтение	то же для канала 7
89h	запись	Задание номера страницы для канала 6
8Bh	запись	то же для канала 5
8Ah	запись	то же для канала 7
8Fh	запись	то же для канала 4

В следующей таблице приведено описание назначения портов второго контроллера ПДП с указанием аналогичных портов первого контроллера. Форматы данных, вводимых/выводимых через эти порты совпадают с форматами данных аналогичных портов первого контроллера. При работе с этими портами следует только учитывать, что нумерация каналов начинается не с 0, как у первого контроллера, а с 4 (канал 4 - это канал 0 второго контроллера, канал 5 - это его канал 1 и т. д.).

Порт ПДП-2	Режим	Назначение	Аналог. порт ПДП-1
0D0h	запись	Запись регистра команд ПДП	8h
0D0h	чтение	Чтение регистра состояния ПДП	8h
0D2h	запись	Запись в регистр запросов ПДП	9h
0D4h	запись	Запись бита маски для одного из каналов ПДП	0Ah
0D6h	запись	Запись регистра режимов для одного из каналов ПДП	0Bh
0D8h	запись	Очистка (сброс) триггера-защелки (триггера первого/последнего)	0Ch
0DAh	запись	Программный сброс контроллера	0Dh
0DCh	запись	Очистка битов масок всех каналов	0Eh
0DEh	запись	Запись регистра масок для всех 4-х каналов	0Fh
0DAh	чтение	Чтение рабочего регистра ПДП	0Dh

3. Программируемый контроллер прерываний

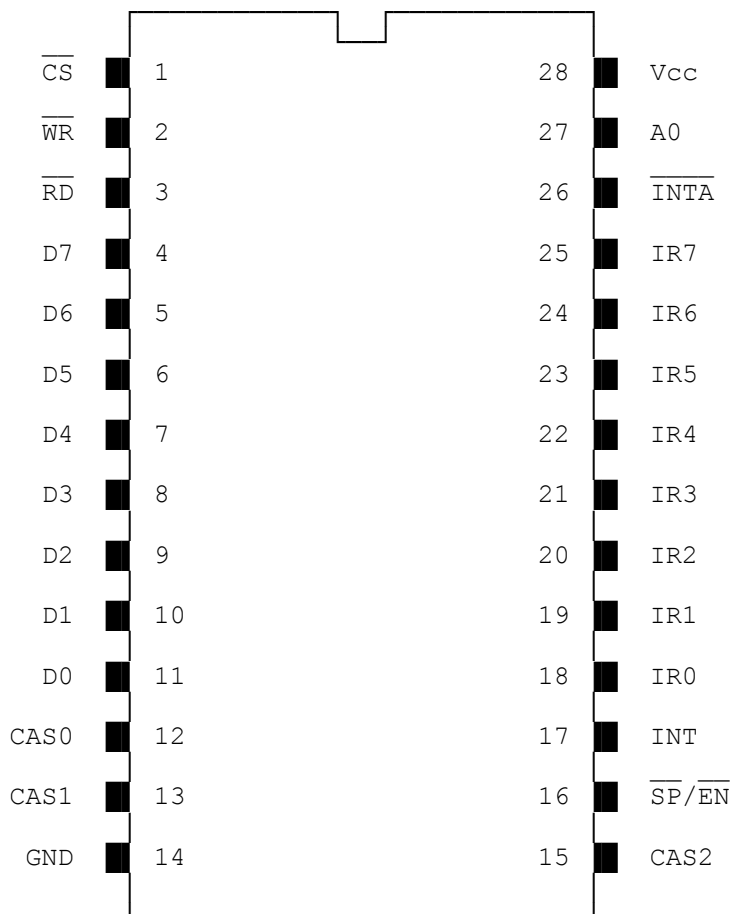
Программируемый контроллер прерываний (ПКП, Programmable Interrupt Controller, PIC) реализует векторную систему

прерываний. Микросхема 8259А фирмы Intel (советский аналог КР580ВН59), а так же ее модификации 8259А-2 и 8259А-8, поддерживает 8 уровней прерываний от восьми различных устройств. Основные функции контроллера:

- фиксация запросов на прерывания от восьми внешних источников;
- программное маскирование поступающих запросов;
- присвоение фиксированных или циклически изменяемых приоритетов входам контроллера, на которые поступают запросы;
- инициация вызова процедуры обработки поступившего аппаратного прерывания.

Количество обслуживаемых внешних источников прерываний может быть увеличено путем каскадирования нескольких контроллеров.

Расположение выводов микросхемы контроллера 8259А приведено на рисунке.



Обозначение	Тип вывода	Назначение
Vcc	-	Питание + 5 В
GND	-	Общий (земля)
$\overline{\text{CS}}$	вход	Выбор микросхемы
$\overline{\text{RD}}$	вход	Чтение
$\overline{\text{WR}}$	вход	Запись

D0-D7	двунаправленные выходы	Выводы на шину данных
CAS0-CAS2	входы/ выходы	Для ведущего контроллера являются выходами, для всех ведомых входами. Предназначены для передачи ведущим информации для выбора необходимого ведомого.
$\overline{SP}/\overline{EN}$	вход	Выбор ведущего контроллера
INT	выход	Запрос на прерывания
\overline{INTA}	вход	Подтверждение прерывания
A0	вход	Выбор регистров контроллера
IR0-IR7	входы	Входы запросов на прерывания от внешних источников.

В состав контроллера входят:

- схема управления чтением/записью;
- схема управления;
- схема каскадирования;
- регистр запросов на прерывания;
- схема обработки приоритетов;
- регистр состояния;
- регистр маскирования запросов на прерывания.

ПКП может находиться в двух основных состояниях: настройки и обслуживания запросов на прерывания. В состоянии настройки контроллер принимает управляющие слова инициализации (Initialization Command Words, ICW), в состоянии обслуживания - операционные управляющие слова (Operation Control Words, OCW). Возможны несколько режимов обслуживания источников прерываний:

- режим фиксированных приоритетов по уровням прерываний;
- два различных варианта циклического сдвига приоритетов;
- режим автоматического завершения обработки прерывания;
- режим специального маскирования;
- режим опроса устройств.

3.1. Описание основных элементов ПКП

Схема управления чтением/записью (Read/Write Control Logic). Основной функцией этого блока является прием команд от микропроцессора и передача ему информации о состоянии ПКП. Обмен с микропроцессором осуществляется через специальный 8-разрядный буфер данных (Data Bus Buffer), являющийся интерфейсом между ПКП и шиной данных. В состав блока входят регистры управляющих слов ICW и OCW. Схема управляется входами CS, RD, WR и A0. Вход CS (Chip select) отвечает за выбор микросхемы. Низкий уровень сигнала на входе CS разрешает выполнение обмена с ПКП. Низкий уровень сигнала на входе WR (Write) разрешает микропроцессору выводить управляющие слова ICW и OCW для приема их ПКП. Низкий уровень сигнала на входе RD (Read) разрешает ПКП передать микропроцессору информацию о состоянии специальных регистров IRR, ISR и IMR, которые описаны ниже.

Все управляющие слова ICW и OCW принимаются контроллером в

виде 9-разрядных значений. Разряды 0 - 7 передаются через 8-разрядный буфер данных. Старший разряд (восьмой, считая с нуля) носит название A0 и устанавливается в 0 или 1 в зависимости от того, через какой из двух возможных портов ввода-вывода (четный или нечетный) было передано управляющее слово. Если для вывода значения использовался порт с четным адресом, A0 будет равен 0, если использовался порт с нечетным адресом на единицу большим, чем предыдущий, тогда A0 будет равен 1.

Регистр запросов на прерывания (Interrupt Request Register, IRR) обслуживается через входы IR0 - IR7 контроллера. Сигнал на одном из входов IR0 - IR7 - это запрос на прерывание соответствующего уровня (0 - 7). В соответствии с сигналом запроса на прерывание схемой управления устанавливается соответствующий бит в регистре IRR.

Регистр состояния (регистр обрабатываемых запросов, In-Service Register, ISR) описывает в битах 0 - 7 прерывания каких уровней (0 - 7) в данный момент обрабатываются.

Регистр маскирования запросов на прерывания (Interrupt Mask Register, IMR) описывает, прерывания каких уровней в настоящий момент замаскированы. Единичное значение бита в IMR указывает на то, что прерывание соответствующего уровня при появлении запроса в IRR блокируется.

Схема обработки приоритетов (шифратор приоритетов, Priority Resolver) определяет, прерывание какого уровня в данный момент является наиболее приоритетным для выполнения.

Схема управления ПКП формирует сигнал запроса на прерывание, поступающий на вход INT (запрос на прерывание) микропроцессора. Если флаг IF регистра флагов процессора равен 1 (прерывания разрешены), процессор отвечает сигналом по линии INTA (подтверждение прерывания), после чего сбрасывается в 0 разряд IRR и устанавливается в 1 разряд ISR, соответствующие уровню обрабатываемого прерывания. После получения второго сигнала подтверждения от процессора по линии INTA, ПКП передает на шину данных 8-битовый номер прерывания. Данная последовательность работы схемы управления выполняется при подключении ПКП к системе с микропроцессорами 8088/8086. При работе с микропроцессорами 8080/8085 последовательность работы схемы управления несколько отличается от описанной выше. Основное отличие состоит в том, что процессору передается не только номер прерывания, но и код команды процессора INT (прерывание) - байт 0CDh.

Схема каскадирования отвечает за работу каскада из нескольких контроллеров. При подключении к ведущему контроллеру выход INT каждого ведомого подключается к одному из входов IR0 - IR7 ведущего. Далее этот сигнал передается ведущим на вход INT процессора. Когда процессор возвращает сигнал INTA, ведущий контроллер не только устанавливает бит в ISR и сбрасывает бит в IRR, но и выдает на свои выходы CAS0 - CAS2 номер уровня прерывания, к которому подключен ведомый, пославший запрос на прерывание. Сигналы по линии CAS0 - CAS2 принимаются всеми ведомыми, однако обрабатываются только тем, который подключен к линии IR с соответствующим номером.

3.2. Режимы работы ПКП

1. Режим фиксированных приоритетов (Fixed Priority, Fully

Nested Mode). В этом режиме контроллер находится сразу после инициализации. Запросы прерываний имеют жесткие приоритеты от 0 до 7 (0 – высший) и обрабатываются в соответствии с приоритетами. Прерывание с меньшим приоритетом никогда не будет обработано, если в процессе обработки прерываний с более высокими приоритетами постоянно возникают запросы на эти прерывания.

2. Автоматический сдвиг приоритетов (Automatic Rotation). В этом режиме дается возможность обработать прерывания всех уровней без их дискриминации. Например, после обработки прерывания уровня 4 ему автоматически присваивается низший приоритет, при этом приоритеты для всех остальных уровней циклически сдвигаются и прерывания уровня 5 будут иметь в данной ситуации высший приоритет и, следовательно, возможность быть обработанными.

3. Программно-управляемый сдвиг приоритетов (Specific Rotation). Программист может сам передать команду циклического сдвига приоритетов ПКП, задав соответствующее управляющее слово. В команде задается номер уровня, которому требуется присвоить максимальный приоритет. После выполнения такой команды устройство работает так же, как и в режиме фиксированных приоритетов, с учетом их сдвига. Приоритеты сдвигаются циклически, таким образом если максимальный приоритет был назначен уровню 3, то уровень 2 получит минимальный и будет обрабатываться последним.

4. Автоматическое завершение обработки прерывания (Automatic End Of Interrupt, AEOI). В обычном режиме работы процедура обработки аппаратного прерывания должна перед своим завершением очистить свой бит в ISR специальной командой, иначе новые прерывания не будут обрабатываться ПКП. В режиме AEOI нужный бит в ISR автоматически сбрасывается в тот момент, когда начинается обработка прерывания нужной процедурой обработки и от нее не требуется издавать команду завершения обработки прерывания (EOI). Сложность работы в данном режиме обуславливается тем, что все процедуры обработки аппаратных прерываний должны быть повторно входимыми, т. к. за время их работы могут повторно возникнуть прерывания того же уровня.

5. Режим специальной маски (Special Mask Mode). Данный режим позволяет отменить приоритетное упорядочение обработки запросов и обрабатывать их по мере поступления. После отмены режима специальной маски предшествующий порядок приоритетов уровней сохраняется.

6. Режим опроса (Polling Mode). В этом режиме аппаратные прерывания не происходят автоматически. Появление запросов на прерывание должно определяться считыванием IRR. Данный режим позволяет так же получить от ПКП информацию о наличии запросов на прерывания и, если запросы имеются, номер уровня с максимальным приоритетом, по которому есть запрос.

3.3. Программирование ПКП

Ниже описывается управление ПКП в ПЭВМ на базе микропроцессоров Intel 8086/8088/80286. Возможности, позволяющие включить ПКП в ПЭВМ, использующие другие процессоры, опущены.

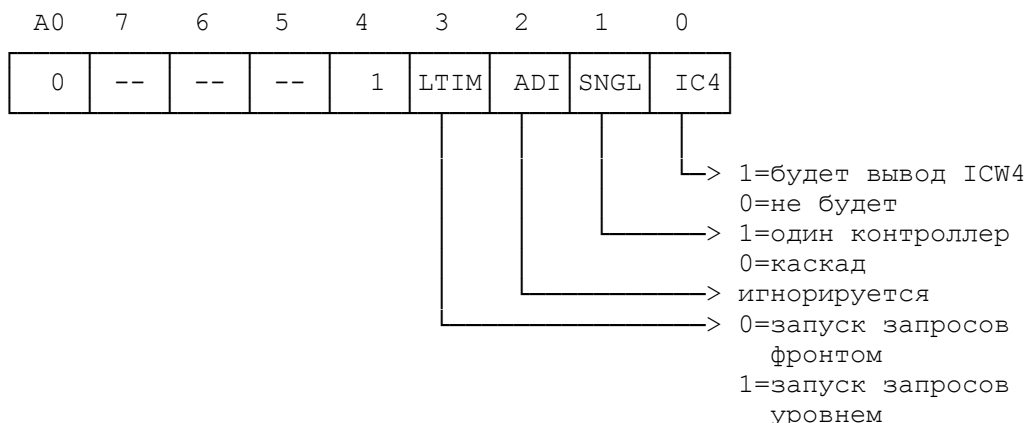
Для вывода информации в ПКП используются 2 порта ввода-вывода. Порт с четным адресом (обычно это порт 20h) и порт с нечетным адресом (обычно 21h). Через эти порты могут быть

переданы 4 слова инициализации (Initialization Control Word, ICW1 - ICW4), задающие режим работы ПКП, и 3 операционных управляющих слова (слова рабочих приказов, Operation Control Words, OCW1 - OCW3). В порт с четным адресом выводятся ICW1, OCW2 и OCW3. OCW2 отличается от OCW3 тем, что бит 3 в ocw2 равен 0, а в OCW3 равен 1. В то же время бит 4 в OCW2 и OCW3 равен 0, а в ICW1 равен 1. Таким образом по значению, выводимому в порт с четным адресом, однозначно определяется, в какой регистр (ICW1, OCW2 или OCW3) заносятся данные.

Порт с нечетным адресом используется для вывода ICW2, ICW3, ICW4 и OCW1. Неоднозначности интерпретации данных в этом случае так же не возникает, т. к. слова инициализации ICW2 - ICW4 должны непосредственно следовать за ICW1, выведенным в порт с четным адресом и выводить в промежутке между ними OCW1 не следует, оно не будет опознано контроллером.

Выводом в порт с четным адресом управляющего слова инициализации ICW1 начинается инициализация ПКП. В процессе инициализации контроллер последовательно принимает управляющие слова ICW1 - ICW4. При наличии в системе одного контроллера ICW3 не выводится. Наличие ICW4 определяется содержанием ICW1. При наличии каскада из нескольких ПКП каждый из них инициализируется отдельно.

Формат ICW1 следующий:



Бит 4, равный 1, определяет, что выводится ICW1, а не OCW2 или OCW3.

Бит 3 (LTIM), равный 0, задает запуск запросов фронтон, при этом действует описанная выше схема: бит в IRR сбрасывается при установке соответствующего бита в ISR.

Бит 2 (ADI) используется только в ПЭВМ на базе микропроцессоров Intel 8080/8085.

Бит 1 (SNGL) указывает на наличие в системе одного контроллера (если равен 1) или каскада из нескольких контроллеров (если равен 0). Если этот бит равен 1, то ICW3 не выводится в процессе инициализации и следом за ICW2 сразу должно следовать ICW4.

Бит 0 (IC4) определяет, будет ли выводиться ICW4. Если ICW4 не выводится (бит 0 = 0), то оно автоматически заполняется нулями. В описываемых ПЭВМ наличие ICW4 обязательно, т. к. тип микропроцессора 8086/8088 задается в нем значением 1 в одном из битов.

Управляющее слово ICW2 задает номер вектора прерывания для

прерываний уровня 0 (например 8 для IBM PC, у которых по уровню 0 происходят прерывания от таймера). Так как вектора аппаратных прерываний располагаются подряд друг за другом, вывод в ICW2 значения 8 не только задает восьмой вектор для таймера, но и девятый для прерываний уровня 1, десятый (0Ah) для прерываний уровня 2 и т. д.

Управляющее слово ICW3 выводится только при наличии каскада и имеет разный формат для ведущего и ведомых контроллеров. ICW3 ведущего указывает, к каким входам IR0 - IR7 подключены ведомые контроллеры, при этом соответствующие биты устанавливаются в 1. Остальные биты при этом равны 0. ICW3 следующего вида:

A0	7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	1	0

задает, что в каскаде имеется 2 ведомых контроллера, подключенных к входам IR1 и IR4.

ICW3 ведомого (подчиненного, slave) ПКП в трех младших битах задает номер уровня, на котором работает ведомый контроллер. Для ведомого контроллера, работающего на уровне 1 ICW3 будет выглядеть следующим образом:

A0	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	1

Если ведомый контроллер работает на уровне 4, то его ICW3 будет таким:

A0	7	6	5	4	3	2	1	0
1	0	0	0	0	0	1	0	0

Последнее слово инициализации (ICW4) имеет следующий формат:

A0	7	6	5	4	3	2	1	0
1	0	0	0	SFNM	BUF	M/S	AEOI	mPM

- > 1=8088/8086
0=8080/8085
- > 1=AEOI
0=EOI
- > 1=ведущий
0=ведомый
(только при BUF=1)
- > 1=режим
буферизации
- > 1=специальный
вложенный режим

Бит 0 (mPM) определяет, с каким микропроцессором работает ПКП (0 - 8080/8085, 1 - 8086/8088).

Бит 1 (AEOI), равный 1, задает режим автоматического завершения обработки прерывания, описанный выше. Если этот бит

равен 0, действует обычное соглашение: процедура обработки аппаратного прерывания должна сама сбрасывать свой бит в ISR.

Бит 2 (M/S) игнорируется, если бит 3 (BUF) = 0. При наличии одного контроллера и BUF = 1 устанавливается в 1. При наличии каскада должен быть равен 1 только для ведущего контроллера.

Бит 4 (SFNM) устанавливает специальный вложенный режим, применяемый при каскадировании для определения приоритетов запросов от разных контроллеров (Special Fully Nested Mode).

После инициализации ПКП готов к работе в заданном режиме. Для изменения режимов работы, задаваемых при инициализации требуется переинициализировать его заново.

В процессе работы с ПКП Вы можете без переинициализации:

- маскировать и размаскировать аппаратные прерывания;
- изменять приоритеты уровней;
- издавать команду завершения обработки аппаратного прерывания;
- устанавливать/сбрасывать режим специальной маски;
- переводить контроллер в режим опроса и считывать состояние регистров ISR и IRR; для этого Вам потребуется вывести в порты ПКП одно из трех слов рабочих приказов OCW1 - OCW3.

Формат первого слова рабочих приказов OCW1:

A0	7	6	5	4	3	2	1	0
1	M7	M6	M5	M4	M3	M2	M1	M0

Единичное значение одного из битов M0 - M7 означает, что прерывания соответствующего уровня (IR0 - IR7) маскируются и не будут обрабатываться контроллером.

Второе слово рабочих приказов (OCW2) предназначено для вывода команды завершения обработки аппаратного прерывания (EOI), циклического сдвига и явного изменения приоритетов уровней. Назначение битов OCW2 следующее:



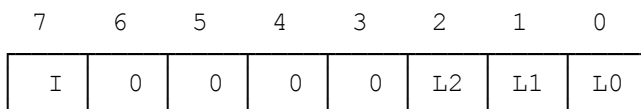
Как уже говорилось, процедура обработки аппаратного

прерывания должна перед своим завершением очистить свой бит в ISR выводом команды завершения обработки прерывания (End Of Interrupt, EOI). Существует два варианта команды EOI: обычный и специфицированный EOI. Обычный EOI очищает бит в ISR, соответствующий прерыванию с максимальным приоритетом. Специфицированный EOI (R=0, SL=1, EOI=1, L0 - L2 равно номеру уровня прерывания) очищает в ISR бит, соответствующий прерыванию с номером, указанным в L0 - L2 независимо от его приоритета. Команды с битом R=1 позволяют изменить приоритеты уровней. Циклический сдвиг приоритетов сдвигает приоритеты влево на единицу, при этом, если после обычного распределения приоритетов, издать команду циклического сдвига, уровень 0 получит низший приоритет, уровень 1 - наивысший, уровень 2 - следующий за ним и т.д. Команда явного назначения низшего приоритета одному из уровней изменяет приоритеты остальных уровней циклически. Таким образом, если Вы зададите низший приоритет уровню 5, то уровень 6 получит наивысший.

Третье слово рабочих приказов OCW3 позволяет установить и отменить режим специальной маски, перевести контроллер в режим опроса и прочитать содержимое IRR и ISR. Назначение битов OCW3 приведено на рисунке



Единичное значение бита P (бит опроса, Polling Bit) переводит контроллер в режим опроса. Если после этого считать данные из порта с четным адресом, в регистр AL загрузится байт следующего содержания:



Если I = 1, значит имеются запросы на прерывания и тогда L0 - L2 - это номер уровня с наивысшим приоритетом, по которому имеется запрос на прерывание.

Если P=0, Вы можете считать информацию из ISR или IRR. Для этого необходимо издать команду чтения ISR или IRR (см. рисунок выше) и затем считать значение из порта с нечетным адресом.

Единичное значение бита ESMM позволяет в зависимости от значения бита SMM установить или отменить режим специальной маски.

4. Таймер и генерация звука

4.1. Программируемый таймер 8253

Для задания временных интервалов и формирования сигналов с

различными временными параметрами в IBM PC/XT применяется программируемый таймер 8253 (отечественный аналог КР580ВН53), в AT – 8254. С точки зрения программиста они идентичны. В состав таймера входят: буфер шины данных, схема управления вводом-выводом и три независимых канала, каждый из которых содержит регистр режима, схему управления каналом, буфер и 16-разрядный счетчик.

Программирование канала осуществляется путем вывода управляющих слов в регистр режима каналов и начального значения в его счетчики. Каждый канал имеет управляющий вход GATE и выход OUT и может работать в одном из следующих шести режимов.

Режим 0 (прерывание терминального счета). После записи управляющего слова в регистр режима канала на выходе OUT устанавливается напряжение низкого уровня; загрузка счетчика не изменяет это состояние. Затем начинается декремент счетчика (последовательное вычитание из него единицы). В момент, когда счетчик обнулится, на выходе OUT устанавливается напряжение высокого уровня и сохраняется до загрузки счетчика новым значением. Счет возможен только при наличии сигнала высокого уровня на входе GATE. Низкий уровень этого сигнала или ниспадающий фронт запрещают счет.

Перезагрузка счетчика во время приводит к следующему: загрузка младшего байта останавливает текущий счет, загрузка старшего байта запускает новый цикл счета. Минимально допустимое значение счетчика равно 2.

Режим 1 (ждущий мультивибратор). На выходе OUT формируется отрицательный импульс длительностью $t=n*T$, где n – число, загруженное в счетчик, T – период тактовых импульсов. Низкий уровень на выходе OUT устанавливается со следующего такта после подачи на вход GATE сигнала высокого уровня.

Загрузка в счетчик нового числа не влияет на длительность текущего импульса, а учитывается при следующем запуске. Перезапуск счетчика производится нарастающим фронтом входа GATE (без перезагрузки счетчика). Минимальное допустимое $n=1$.

Режим 2 (генератор частоты). Каждый раз после достижения счетчиком нуля на выходе OUT появляется отрицательный импульс с длительностью один такт. Перезагрузка счетчика сказывается только после перезапуска счетчика. При исчезновении сигнала высокого уровня на входе GATE прекращается счет и на выход OUT подается напряжение высокого уровня. Перезапуск счетчика происходит при наличии на входе GATE сигнала высокого уровня.

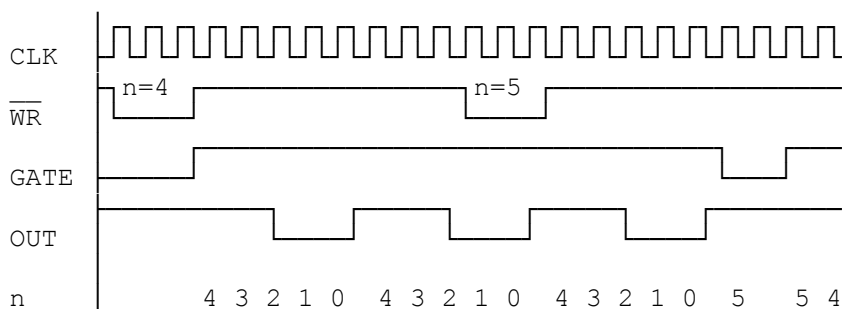
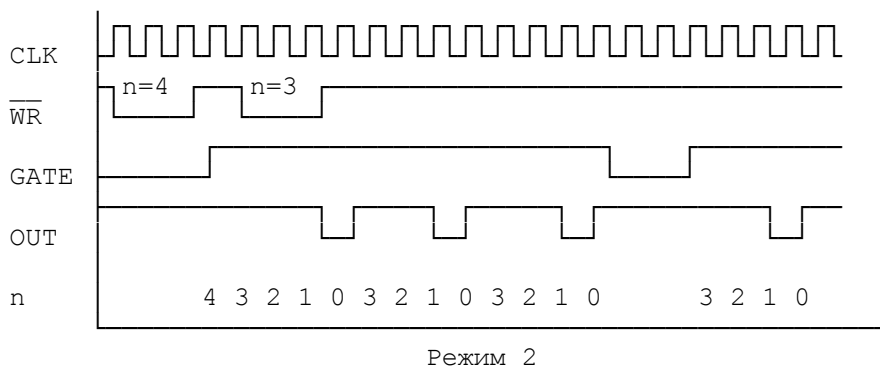
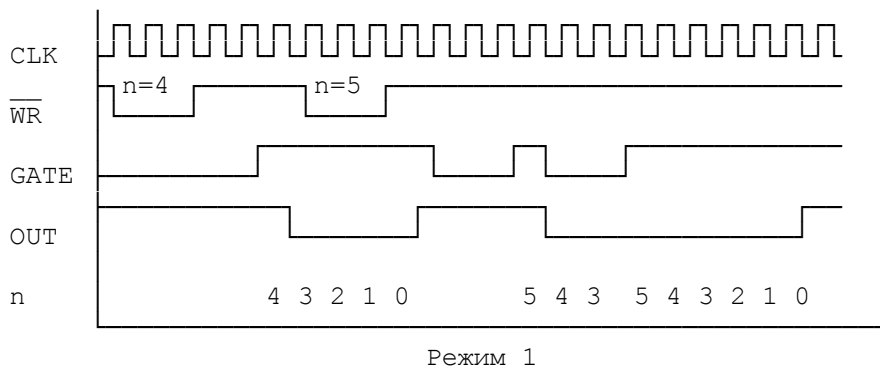
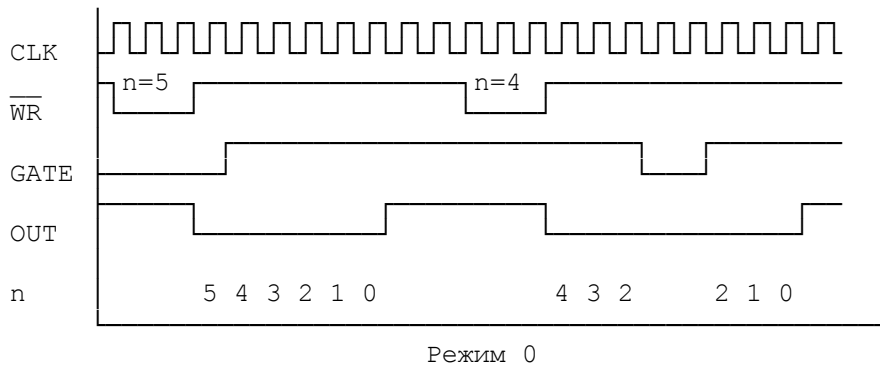
Режим 3 (генератор меандра). Аналогичен режиму 2, но положительный уровень выходного сигнала занимает первый полупериод, а отрицательный – второй полупериод. Точнее, если n (начальное значение счетчика) четно, то длительность положительного и отрицательного полупериодов равна $n*T/2$; если же n нечетно – то $(n+1)*T/2$ и $(n-1)*T/2$ соответственно. Низкий уровень сигнала на входе GATE запрещает счет, на выходе OUT устанавливается сигнал высокого уровня. Высокий уровень GATE разрешает счет, а нарастание его запускает счетчик начального состояния. Отметим, что $n=3$ в этом режиме недопустимо.

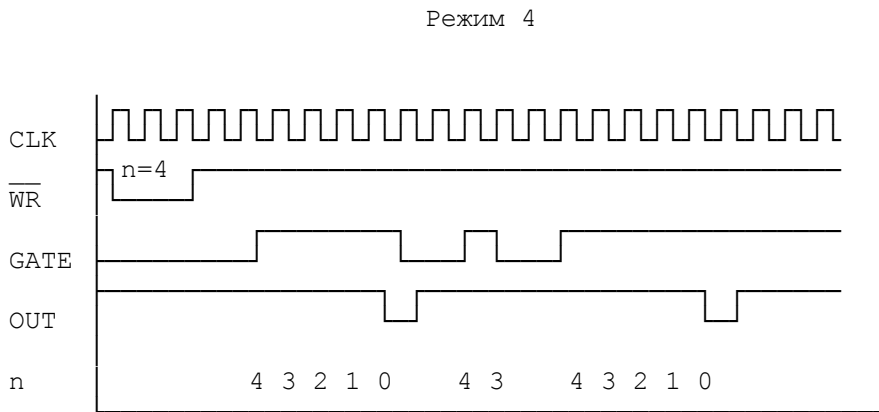
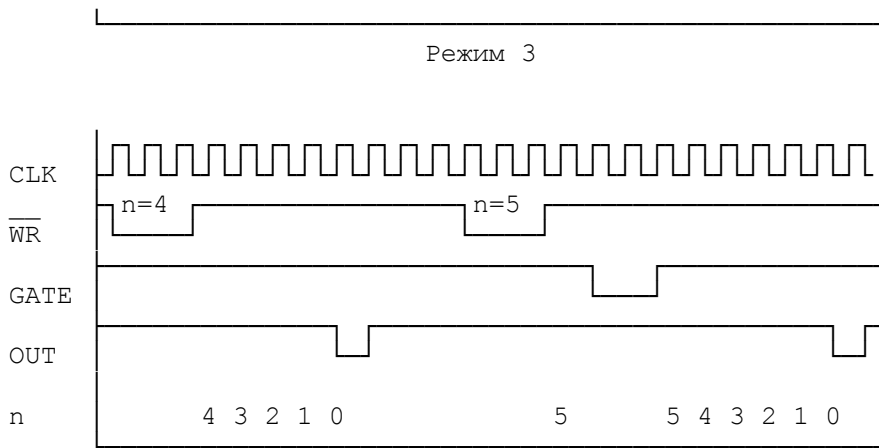
Режим 4 (счетчик событий). По окончании отсчета числа, загруженного в счетчик, на выходе OUT формируется отрицательный импульс длительностью один такт. Запись в счетчик во время счета младшего байта не влияет на текущий счет, а запись старшего байта перезапускает счетчик. Низкий уровень входа GATE запрещает

счет, высокий - разрешает. Минимальное допустимое значение счетчика равно 1.

Режим 5 (счетчик событий с автозагрузкой). Отличие от режима 4 состоит в том, что каждое нарастание сигнала на входе GATE перезапускает счетчик. Перезагрузка счетчика не влияет на текущий цикл, однако следующий цикл определяется вновь занесенным числом.

Временные диаграммы режимов таймера приведены на следующем рисунке:



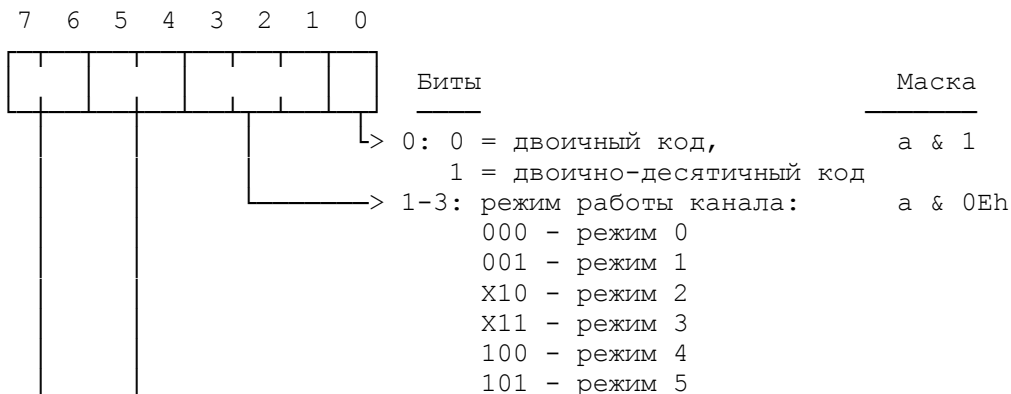


Режим 5

В IBM PC таймер имеет базовый адрес 40h и следующие программируемые регистры:

Адрес	Операция	Назначение
40h	запись чтение	Загрузка счетчика канала 0 Чтение счетчика канала 0
41h	запись чтение	Загрузка счетчика канала 1 Чтение счетчика канала 1
42h	запись чтение	Загрузка счетчика канала 2 Чтение счетчика канала 2
43h	запись	Запись управляющего слова в регистр режима канала

Управляющее слово имеет следующий формат:



┌	└───>	4-5: вид загрузки счетчика: a & 30h
		00 - "защелкивание" (биты 0-3 безразличны)
		01 - только младший байт
		10 - только старший байт
		00 - младший байт, затем старший
└	───>	6-7: номер канала: a & C0h
		00 - канал 0
		01 - канал 1
		10 - канал 2
		11 - запрещенная комбинация

Существует два способа чтения текущего значения счетчика канала.

1. Чтение с остановом счетчика. Для обеспечения стабильных показаний необходимо приостановить работу канала либо подачей сигнала низкого уровня на вход GATE (кроме режима 1), либо блокированием тактовых импульсов.

2. Чтение "на лету". Для считывания счетчика без остановки процесса счета используется посылка в порт 43h управляющего слова в режиме "защелкивания" (см. выше). Это управляющее слово фиксирует текущее значение счетчика и Вы можете считать его младший байт, а затем старший байт.

4.2. Таймер на системной плате IBM PC

В IBM PC каналы таймера имеют следующее назначение.

Канал	Назначение	Режим
0	системные часы (IRQ0)	3, счетчик=0 (65536)
1	запрос для канала 0 ПДП (регенерация памяти)	2, счетчик=18
2	генератор звука	—

Тактовая частота каждого канала равна 1,19318 МГц, т. е. каждый такт имеет длительность 0,84 мсек. Вход GATE каналов 0 и 1 всегда имеют высокий уровень, поэтому счет на этих каналах разрешен всегда. Вход GATE канала 2 управляется битом 0 порта PB интерфейса 8255 (см. гл. 5), связанного с портом 61h.

При начальной загрузке BIOS инициализирует канал 0 для работы в режиме 3 со счетчиком 0 (т. е. 65536 декрементов на цикл счета). Поэтому частота системных часов равна $1,19 \text{ МГц} / 65536 = 18,2 \text{ Гц}$ и прерывание IRQ0, связанное с вектором Int 8, происходит 18,2 раз в секунду, т. е. каждые 55 мсек.

Вы можете перепрограммировать канал 0, но тогда

- BIOS не сможет отслеживать текущее время и дату;

- нарушится работа с гибкими дисками, т. к. включение и выключение их двигателей отсчитывается по текущему времени.

Канал 1 работает в режиме 2 со счетчиком 18, поэтому регенерация памяти происходит каждые 18 мсек. Перепрограммировать его нельзя, т. к. это приведет к потере данных в ОЗУ.

Программирование канала 2 описано в следующем пункте.

4.3. Генерация звука

На вход звукогенератора поступает логическое "И" двух сигналов: выхода OUT 2-го канала таймера и содержимого бита 1 порта PB интерфейса 8255. Поэтому простейший способ генерации звука состоит в программировании канала 2 таймера так, чтобы он выдавал прямоугольный импульс заданной частоты, лежащий в звуковом диапазоне (20 Гц - 20 КГц). Для этого следует использовать режим таймера 3 с подходящим начальным значением счетчика. Если затем установить биты 0 и 1 порта PB, то импульс начнет поступать на вход звукогенератора (бит 0 - это вход GATE канала 2, разрешающий счет, а бит 1 - разрешение выдачи выхода OUT на вход звукогенератора). Для выключения звука достаточно сбросить биты 0 - 1 в PB. Преимущество этого метода состоит в том, что, запустив генерацию звука, ЦП может выполнять другие действия. Значение счетчика 2-го канала вычисляется по формуле $n=1193181/f=1234DDh/f$ (1193181 - тактовая частота таймера в Гц, f - требуемая частота звука).

Пример:

```

;
; подпрограмма генерации звука
; Вход: AX= частота звука в Гц
;
Sound proc near
    push    ax           ;сохранить регистры
    push    bx
    push    dx
    mov     bx,ax        ;частота
    mov     ax,34DDh
    mov     dx,12h      ;(dx,ax)=1193181
    cmp     dx,bx       ;если bx < 18Гц, то выход
    jnb     Done        ;чтобы избежать переполнения
    div     bx           ;ax=(dx,ax)/bx
    mov     bx,ax       ;счетчик таймера
    in     al,61h       ;порт PB
    or     al,3         ;установить биты 0-1
    out    61h,al
    mov     al,00001011b ;управляющее слово таймера:
                        ;канал 2, режим 3, двоичное слово

    mov     dx,43h
    out    dx,al        ;вывод в регистр режима
    dec     dx
    mov     al,b1
    out    dx,al        ;младший байт счетчика
    mov     al,bh
    out    dx,al        ;старший байт счетчика
Done:
    pop     dx          ;восстановить регистры
    pop     bx
    pop     ax
    ret
Sound endp
;
; подпрограмма выключения звука
;
No_Sound proc near
    push    ax
    in     al,61h       ;порт PB
    and    al,not 3     ;сброс битов 0-1
```

```

        out    61h,al
        pop    ax
        ret
No_Sound endp

```

Для генерации музыки можно использовать следующий алгоритм. Пусть octave - номер октавы (1, 2, ...); note - номер ноты в октаве (до=1, до#=2, ..., си=12). Тогда:

```

f := 32.625;
for i := 1 to octave do f := 2*f;
for i := 1 to note do f := 1.059463094*f
Sound (round(f));

```

Второй способ генерации звука состоит в том, чтобы генерировать звуковые импульсы не выходом таймера, а установкой и сбросом бита 1 в регистре PB. Для управления задержками можно использовать программный цикл, например:

```

;
; Программа генерации звука
; Вход: N=количество импульсов
;       FREQ=длительность импульса
;
in      al,61h      ;порт PB
and     al,not      ;сброс бита 0
mov     dx,N
Repeat:
or      al,2
out     61h,al      ;установить бит 1
mov     cx,FREQ
Wait_On:
loop    Wait_On
and     al,not 2    ;сбросить бит 1
out     61h,al
mov     cx,FREQ
Wait_Off:
loop    Wait_Off
dec     dx
jnz     Repeat

```

Программирование этого способа основано на следующих формулах. Если T - длительность звучания в сек, f - частота звука в Гц, а t - тактовая частота процессора ПЭВМ, то

$$N=T*f; \text{ FREQ}=t/(34*f).$$

Константа 34 объясняется тем, что длительность каждого из циклов Wait_on и Wait_off приблизительно равна (17*cx) машинных тактов.

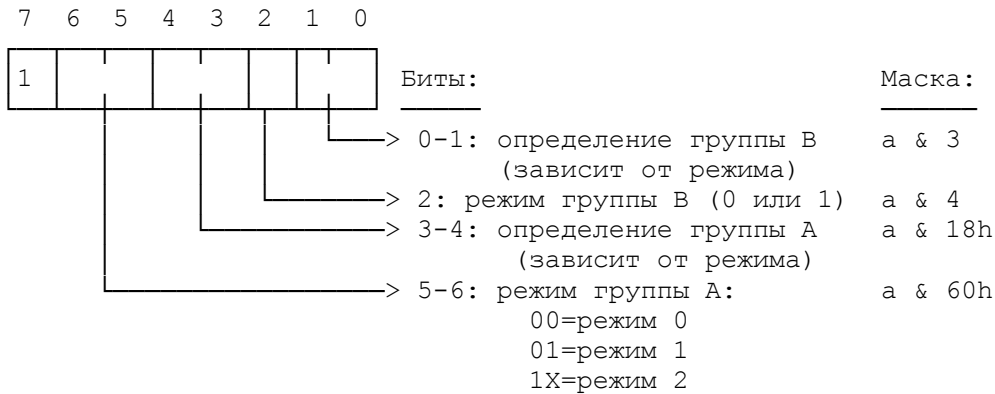
Неудобство этого метода состоит в том, что ЦП полностью занят генерацией звука - любое прерывание испортит временную диаграмму звукового сигнала. Однако он обеспечивает точную подстройку частоты и позволяет создавать несимметричные импульсы варьированием задержек в циклах Wait_on и Wait_off.

5. Программируемый периферийный интерфейс

5.1. Описание ППИ 8255

В состав IBM PC/XT входит программируемый периферийный интерфейс (ППИ) 8255 (отечественный аналог КР580ИК55). ППИ

содержит три 8-разрядных порта (РА, РВ и РС) для организации обмена между ЦП и внешними устройствами и регистр управления. Порты сгруппированы в две группы: группа А состоит из РА и старшей тетрады РС, группа В - из РВ и младшей тетрады РС. Группа А может работать в трех режимах (0, 1, 2); группа В - в двух (0 и 1). Режимы определяются содержимым регистра управления, имеющим следующий формат:



Режим 0. В этом режиме группа разделяется на два набора (порт и тетрада РС). Каждый набор может использоваться только для ввода. Биты определения группы в регистре управления определяют, какие наборы предназначены для ввода, а какие - для вывода (бит, равный нулю, задает набор для вывода, единица - для ввода):

Бит	Набор
4	РА
3	Старшая тетрада РС
1	РВ
0	Младшая тетрада РС

Режим 1. Порты РА и РВ используются для передачи данных, РС - для управляющих сигналов. Дальнейшее описание сделано на примере группы А. Бит 4 регистра управления задает использование порта РА (1 = порт ввода, 0 = порт вывода).

При вводе биты порта РС имеют следующее значение:

Бит (ы)	Назначение
4 (STBA)	0 = "стробировать" шину данных в порт РА
5 (IBFA)	1 = РА содержит данные, еще не выбранные процессором
6-7	если бит 3 регистра управления равен 0, то это линии управления периферийным устройством; если он равен 1, то это линии ввода состояния устройства.

При выводе:

Бит (ы)	Назначение
4-5	то же, что 6,7 при вводе
6 (OBFA)	1 = РА содержит данные, еще не переданные периферийному устройству
7 (ASKA)	0 = данные из РА приняты устройством

С группой А ассоциируется бит 3 порта РС, обозначаемый INTRA. Этот сигнал применяется как линия запроса прерывания системной шины. При вводе в порт РА на этом выходе формируется 1, когда новые данные помещаются в РА, и 0, когда процессор считывает данные. При выводе INTRA = 1, когда содержимое РА принято устройством, и 0, когда ЦП загружает новые данные. Разрешением прерываний управляют бит 4 РС в режиме ввода и бит 6 в режиме вывода (1 = разрешить прерывание).

Для группы В ввод или вывод задается битом 1 регистра управления. При вводе биты 2 и 1 порта РС именуется STBV и IBFB (их назначение аналогично битам STVA и IBFA). При выводе биты 2 и 1 именуется OBFV и ASKB (см. выше). Бит 0 становится INTRV, разрешениями прерываний управляет бит 2.

Режим 2. Этот режим допустим только для группы А. В нем порт А становится двунаправленным, а старшая тетрада РС определяется так:

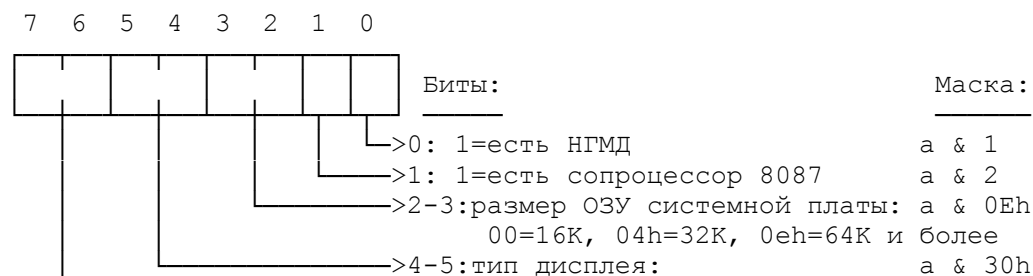
Бит (ы)	Назначение
4 (STVA)	0 = "стробировать" шину данных РА
5 (IBFA)	1 = порт РА загружаются данные с шины, 0 = процессор читает данные
6 (АСКА)	1 = устройство готово читать данные с шины
7 (OBFA)	1 = процессор загружает в порт РА данные, 0 = данные приняты устройством

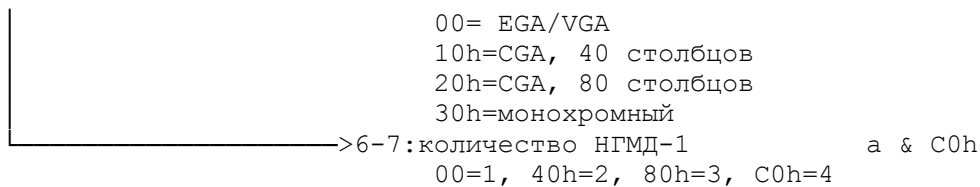
5.2.Использование ППИ в IBM PC

ППИ в IBM PC/XT занимает порты 60h - 63h. BIOS заносит в регистр управления (порт 63h) 99h, т. е. программирует группы А и В в режим 0, причем РА и РС является портами ввода, а РС - портом вывода:

Адрес	Операция	Назначение
60h	чтение	Ввод из порта РА
61h	запись чтение	Ввод в порт РВ Чтение последнего вывода в порт РВ
62h	чтение	Вывод из порта РС
63h	чтение/ запись	Регистр управления ППИ

Содержимое порта РА: если бит 7 в РВ равен 0, то чтение из этого порта означает ввод скан-кода последней нажатой клавиши (см. гл. 6); если же бит 7 в РВ установлен, то РА содержит положение DIP-переключателей колодки SW1 системной платы:

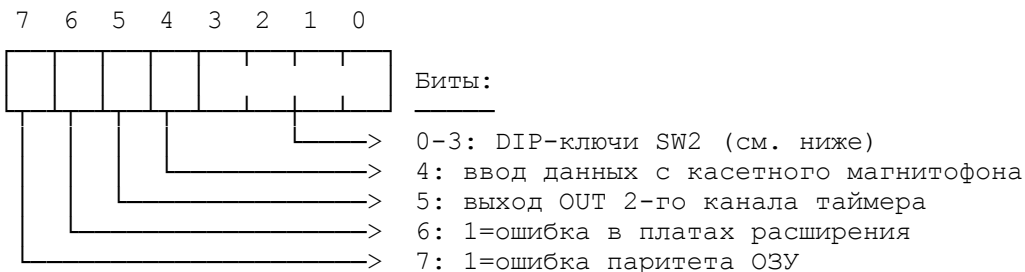




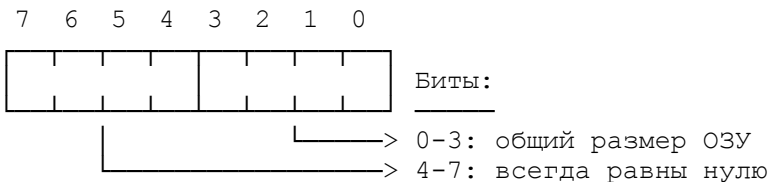
Содержимое порта PB:



Содержимое порта PC зависит от бита 7 порта PB. Если он установлен, то биты 0-3 содержат младшую тетраду колодки DIP-переключателей SW2, если сброшен, то старшую:



Колодка SW2 содержит следующие ключи:



6. Клавиатура

6.1. Клавиатура IBM PC/XT

Связь с клавиатурой осуществляется через порты ППИ 60h и 61h (см. 5.2). При нажатии клавиши происходит аппаратное прерывание IRQ1, связанное с вектором Int 9. Если Вы хотите обрабатывать это прерывание сами, то программа обработки прерывания должна иметь следующий вид:

```

Int_09h:
    push    ax
    push    di
    push    es
    in      al,60h    ;скан. код клавиши из PA
    les     di,buffer ;адрес буфера
    stosb                   ;сохранить код в буфере
    pop     es
    pop     di
    in      al,61h    ;ввод порта PB
    mov     ah,al
    or      al,80h    ;установить бит "подтверждения ввода"
    out     61h,al
    xchg    ah,al     ;вывести старое значение PB
    out     61h,al
    mov     al,20h    ;послать сигнал EOI
    out     20h,al    ;контроллеру прерываний
    pop     ax
    iret

```

Перечень скан-кодов всех клавиш приведен в приложении 1. Вы можете запретить прерывания от клавиатуры либо маскировкой бита 1 в регистре IMR контроллера прерываний, либо установкой бита 6 порта PB ППИ:

```

    in      al,21h    ;чтение IMR
    or      al,2      ;установка бита 1
    out     21h,al    ;запись IMR
или
    in      al,61h    ;чтение PB
    or      al,40h    ;установка бита 6
    out     61h,al    ;запись PB

```

Для разрешения работы клавиатуры необходимо выполнить обратные действия.

6.2. Клавиатура IBM PC AT

Взаимодействие с клавиатурой в PC AT базируется на микропроцессоре Intel 8042; благодаря этому программист имеет гораздо большие возможности по управлению клавиатурой, нежели в PC/XT.

Управление клавиатурой производится обращением к порту ввода-вывода 64h. Для совместимости со старыми ПЭВМ допустимо использование порта 60h. Если приведенные ниже примеры не работают на Вашей ПЭВМ, то используйте порт 60h.

Порт 64h предназначен для вывода команд и данных и для чтения состояния клавиатуры. Общая схема посылки команд клавиатуре такова:

```

cli                ;запрет прерываний
call    wait_kbd   ;ожидание готовности клавиатуры
mov     al,cmd_code ;код команды
out     64h,al     ;вывод в порт
sti                ;разрешение прерываний

```

Для команд, сопровождающихся байтом данных, эта схема принимает вид:

```

cli
call    wait_kbd

```



```

mov     al,cmd_code      ;код команды
out     64h,al
call    wait_kbd
mov     al,data_value    ;байт данных
out     64h,al
sti

```

Проверка ожидания готовности клавиатуры может иметь вид:

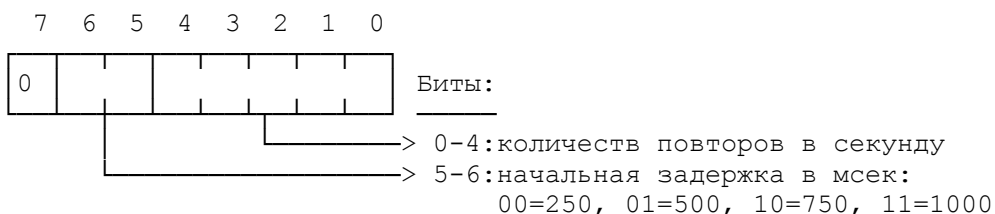
```

wait_kbd:
mov     cx,2500h        ;задержка порядка 10 мсек
test_kbd:
in      al,64h          ;читаем состояние клавиатуры
test    al,2            ;проверка бита готовности
loopnz  test_kbd
ret

```

Авторам известны следующие команды контроллера клавиатуры (этот список, вероятно, не полон и может содержать ошибки, т. к. собран из различных источников):

Код команды	Описание
0Fh	Сброс клавиатуры и запуск внутреннего теста.
0Feh	Повторить последнюю передачу.
0FDh	Вызов прерывания IRQ1.
0F6h	Привести клавиатуру в исходное состояние и разрешить сканирование.
0F5h	Привести клавиатуру в исходное состояние и запретить сканирование.
0F4h	Сбросить буфер клавиатуры и начать сканирование.
0F3h	Задать задержку и скорость ввода символов. Первоначально клавиатура начинает повторять нажатую клавишу через 0,5 сек со скоростью 10 повторов в секунду. Вы можете изменить эти параметры, послав следующий байт данных:

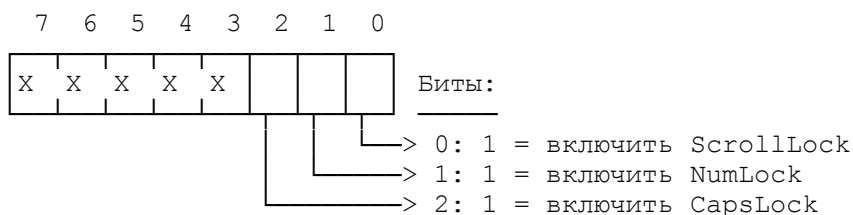


Количество повторов задается следующей таблицей:

Конст.	Скорость	Конст.	Скорость	Конст.	Скорость
00h	30.0	0Bh	10.9	16h	4.3
01h	26.7	0Ch	10.0	17h	4.0
02h	24.0	0Dh	9.2	18h	3.7
03h	21.8	0Eh	8.6	19h	3.3
04h	20.0	0Fh	8.0	1Ah	3.0
05h	18.5	10h	7.5	1Bh	2.7
06h	17.1	11h	6.7	1Ch	2.5
07h	16.0	12h	6.0	1Dh	2.3
08h	15.0	13h	5.5	1Eh	2.1
09h	13.3	14h	5.0	1Fh	2.0
0Ah	12.0	15h	4.6		

0Eeh Эхо (для диагностики). Просто возвращает 0Eeh.

0EDh Управление светодиодами клавиатуры. Для включения или выключения светодиодов пошлите такой байт данных:



0DFh Разрешить 20-битовую адресацию.
0DDh Запретить 20-битовую адресацию.
0D1h Запись в порт вывода 8042.
0D0h Чтение порта вывода 8042.
0C0h Чтение порта ввода 8042.
0AEh Разрешение клавиатуры.
0ADh Запрещение клавиатуры.
0ACh Чтение ОЗУ контроллера.
0ABh Тест синхронизации и данных.
0AAh Внутренний тест клавиатуры.

7. Дисплеи

7.1. Общее описание

Программирование дисплея ПЭВМ IBM PC определяется типом дисплейного адаптера и типом видеомонитора, подключенного к этому адаптеру. В этой главе описаны четыре наиболее распространенных видеоадаптера IBM PC, а именно:

- монохромный графический адаптер MGA;
- цветной графический адаптер CGA;
- расширенный графический адаптер EGA;
- графический видеомассив VGA.

Центральным устройством в управлении видеомонитором является контроллер электронно-лучевой трубки (ЭЛТ). В IBM PC применяются программируемые контроллеры фирмы Motorola, позволяющие задать все основные параметры развертки изображения на экране. Для понимания дальнейшего материала необходимо ввести ряд часто используемых понятий. Режим работы видеомонитора характеризуется следующими понятиями:

- вертикальным разрешением (количество строк раstra на экране);
- горизонтальным разрешением (количество символов или точек в строке);
- представлением данных в видеопамати;
- атрибутами вывода, задающими цвет, мерцание и т. п.

Горизонтальное и вертикальное разрешение определяются последовательностью согласованных во времени сигналов графического адаптера, управляющих движением луча ЭЛТ. Луч ЭЛТ движется с постоянной скоростью вдоль строки раstra и вниз с одной строки раstra на следующую, обновляя (разворачивая) изображение на экране. Для перевода луча из крайней правой точки строки в крайнюю левую точку следующей строки (обратный горизонтальный ход луча) видеоадаптер генерирует сигнал горизонтальной синхронизации; для перемещения луча из крайней правой позиции нижней строки экрана в крайнюю левую позицию

первой строки (вертикальный обратный ход луча) - сигнал вертикальной синхронизации.

Видеоадаптеры всегда программируются так, чтобы вывод видеобуфера на экране проходил быстрее, чем развертка одного кадра. Это дает возможность вывести на экран бордюр (overscan), окаймляющий изображение и позволяющий центрировать его на экране.

Длительность сигналов развертки принято задавать в так называемых символьных тактах (character clock). Символьный такт - это отрезок времени, необходимый для вывода на экран 8 точек в графическом режиме и 8 или 9 точек в алфавитно-цифровых режимах.

Параметры развертки заносятся в определенные регистры контроллера ЭЛТ; назначение этих регистров различно в контроллерах MGA/CGA и EGA/VGA.

7.2. Контроллеры ЭЛТ

Адаптеры MGA и CGA содержат контроллер ЭЛТ Motorola 6845, имеющий следующие программируемые регистры:

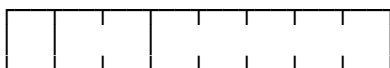
Регистр	Наименование	Описание
R0	Общая длительность строки (horizontal total)	Время вывода строки раstra в символьных тактах.
R1	Ширина кадра (horizontal displayed)	Количество символов в строке.
R2	Позиция горизонтальной синхронизации (horizontal sync position)	Начало горизонтальной синхронизации в символьных тактах.
R3	Ширина синхронизации (sync width)	Длительность синхронизации (младшая тетрада - вертикальной, старшая - горизонтальной).
R4	Размер кадра (vertical total)	Общее количество строк раstra в кадре.
R5	Вертикальное выравнивание (vertical adjust)	Поправка для частоты 50 или 60 Гц.
R6	Высота кадра (vertical displayed)	Количество строк изображения в кадре.
R7	Позиция вертикальной синхронизации (vertical sync position)	Начало вертикальной синхронизации в символьных тактах.
R8	Биты 4-5: режим расслоения (interlace mode); биты 6-7: сдвиг (skew)	
R9	Макс. число строк раstra	Количество строк раstra на строку символов.
R10	Начало курсора	Начальная строка курсора.
R11	Конец курсора	Конечная строка курсора.
R12	\Начальный адрес	Старший байт
R13	/видеопамяти	Младший байт
R14	\Позиция	Старший байт
R15	/курсора	Младший байт
R16	\Позиция	Старший байт
R17	/светового пера	Младший байт

Адаптеры EGA и VGA имеют следующие программируемые регистры, эмулирующие контроллер ЭЛТ:

Регистр	Наименование	Описание
---------	--------------	----------

R0	Общая длительность строки (horizontal total)	Время вывода строки раstra в символьных тактах.
R1	Ширина кадра (horizontal display enable end)	Количество символов в строке (см. R7).
R2	Начало горизонтального гашения (start horizontal blank)	Символьная позиция, в которой начинается горизонтальное гашение луча.
R3	Конец горизонтального гашения (end horizontal blank)	Символьная позиция, в которой заканчивается горизонтальное гашение луча (в битах 0-4); разрешение сдвига (биты 5-6).
R4	Начало обратного хода луча по горизонтали (start horizontal retrace)	Символьная позиция начала горизонтальной синхронизации.
R5	Конец обратного хода луча по горизонтали (end horizontal retrace)	Символьная позиция окончания горизонтальной синхронизации.

7 6 5 4 3 2 1 0



Биты:

- > 0-4: ширина обратного хода луча по горизонтали
- > 5-6: задержка обратного хода в символьных тактах
- > 7: 1=начинать панорамирование с четного адреса, 0=с нечетного

R6	Размер кадра (vertical total)	Общее количество строк раstra в кадре (см. R7).
R7	Регистр переполнения (overflow)	Содержит старшие биты других регистров:

7 6 5 4 3 2 1 0



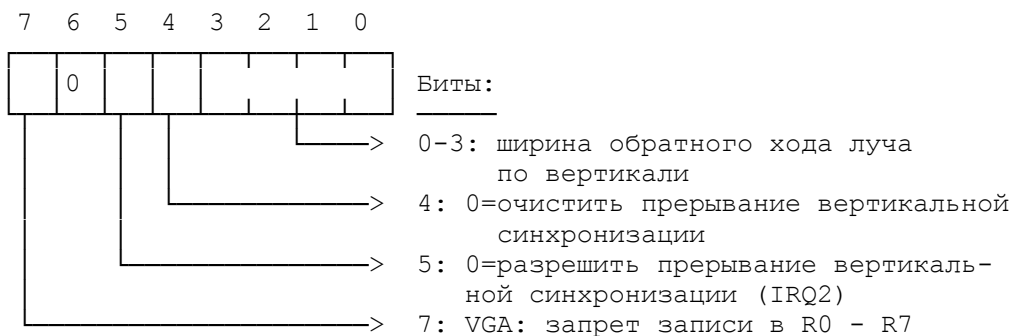
Бит:

- > 0: 8-й бит размера кадра (R6)
- > 1: 8-й бит количества строк отображения в кадре (R18)
- > 2: 8-й бит начала вертикальной синхронизации (R16)
- > 3: 8-й бит начала вертикального гашения луча (R21)
- > 4: 8-й бит сравнения строк (R24)
- > 5: EGA: 8-й бит R10, VGA: 9-й бит R6
- > 6: 9-й бит R18 (только VGA)
- > 7: 9-й бит R16 (только VGA)

R8	Начало раstra (preset row scan)	Первая строка раstra после вертикальной синхронизации.
R9	Макс. число строк раstra	Количество строк раstra на строку символов (в битах 0-4); для VGA: бит 5=9-й бит R21.
R10	Начало курсора	Начальная строка курсора.
R11	Конец курсора	Конечная строка курсора (биты 0-4); сдвиг курсора (биты 5-6).
R12	Начальный адрес	Старший байт
R13	/видеопамяти	Младший байт
R14	Позиция	Старший байт
R15	/курсора	Младший байт

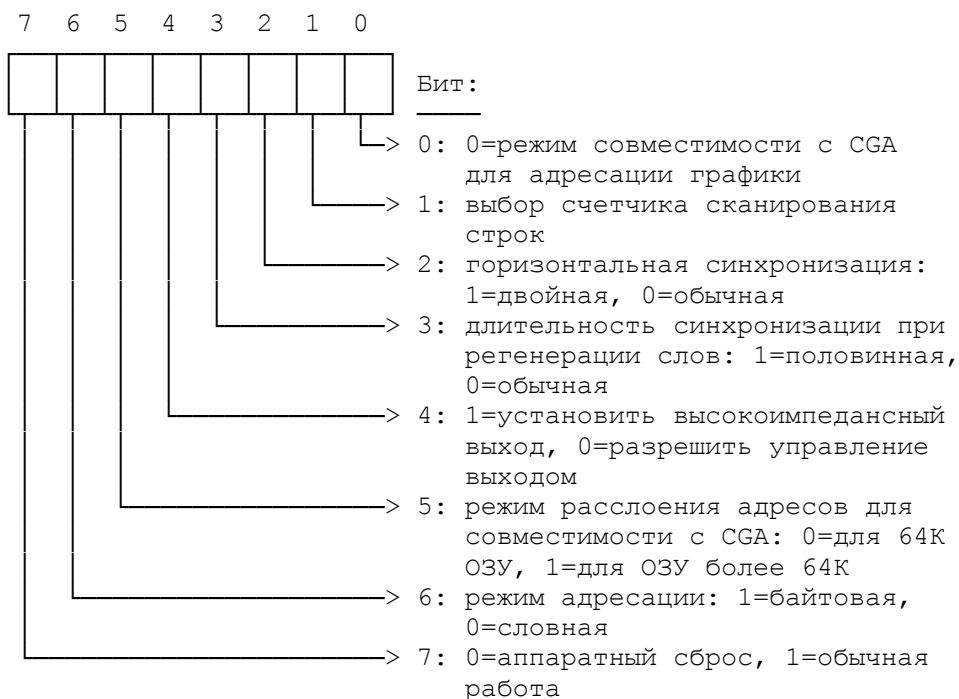
При чтении:

R16\Позиция	Старший байт
R17/светового пера	Младший байт
При записи:	
R16 Начало обратного хода луча по вертикали (start vertical retrace)	Символьная позиция начала вертикальной синхронизации.
R17 Конец обратного хода луча по вертикали (end vertical retrace)	Символьная позиция окончания вертикальной синхронизации.



R18 Количество строк отображения в кадре (vertical displayed end)	Количество строк растра, занятых собственно изображением
R19 Вертикальное смещение (vertical displayed adjustment)	Количество слов данных в логической строке видеопамати.
R20 Позиция подчеркивания	Номер строки растра (биты 0-5).
R21 Начало вертикального гашения (start vertical blank)	Символьная позиция, в которой начинается вертикальное гашение луча.
R22 Конец вертикального гашения (end vertical blank)	Символьная позиция, в которой заканчивается вертикальное гашение луча.

R23 Управление режимами:



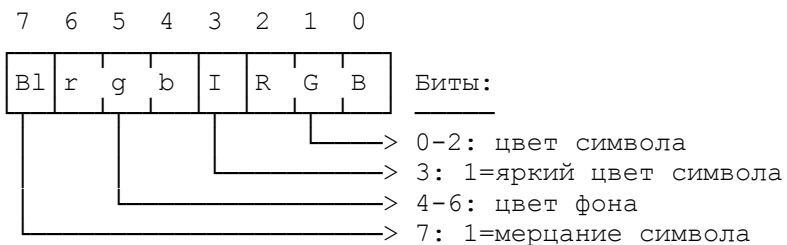
R24 Сравнение строк (line compare)	Строка растра, используемая в операциях расщепления экрана.
------------------------------------	---

В заключение этого раздела подчеркнем, что самостоятельное программирование контроллера ЭЛТ чревато физическим разрушением видеомонитора, поэтому не следует прибегать к нему, если Вы не вполне понимаете принципы расчета параметров развертки. Все необходимые действия по инициализации дисплея обычно выполняются BIOS, поэтому необходимость в непосредственном программировании контроллера ЭЛТ возникает крайне редко.

7.3. Атрибуты символов

В текстовых режимах видеопамять состоит из пар (код символа, атрибут); код символа определяет символ ASCII, выводимый в соответствующей позиции экрана, а атрибут - цвет этого символа. Соответствие цветов атрибутам зависит от типа видеоадаптера.

Байт атрибутов для CGA имеет следующую структуру:



Таким образом, цвет фона может варьироваться от 0 до 7, а цвет символа с учетом яркости - от 0 до 0Fh. Значение атрибута задается формулой:

16 * цвет фона + цвет символа (плюс 128 для мерцания символа)

Цвета задаются следующим образом:

Номер	Цвет	Номер	Цвет
0	Черный	8	Серый
1	Синий	9	Ярко-синий
2	Зеленый	0Ah	Ярко-зеленый
3	Бирюзовый	0Bh	Голубой
4	Красный	0Ch	Алый
5	Сиреневый	0Dh	Розовый
6	Коричневый	0Eh	Желтый
7	Светло-серый	0Fh	Белый

Адаптеры EGA и VGA имеют такую же структуру байта атрибутов, но позволяют переопределять цвета, перечисленные в таблице, перепрограммированием регистров палитр. Поэтому байт атрибутов символа определяет одну из 64 (EGA) или 256 (VGA) возможных комбинаций цветов.

Монохромные дисплеи допускают любые значения байта атрибутов, но реально обрабатывают только следующие его значения:

01h	Подчеркнутый символ
07h	Обычный вывод (светлый символ на темном фоне)
09h	Яркий подчеркнутый символ

0Fh	Ярко-светлый символ на темном фоне
70h	Негатив (темный символ на светлом фоне)
81h	Мерцающий подчеркнутый символ
87h	Мерцающий светлый символ на темном фоне
89h	Мерцающий яркий подчеркнутый символ
8Fh	Мерцающий ярко-светлый символ на темном фоне

Следует учитывать, что старший бит байта атрибутов может означать либо мерцание символа, либо повышенную яркость фона (см. 7.4 - 7.6).

7.4. Монохромный графический адаптер MGA

Монохромный графический адаптер (MGA или Hercules) - это графический видеоконтроллер высокого разрешения, совмещенный с платой печатающего устройства. Для него зарезервированы порты ввода-вывода 3B0h - 3BFh, имеющие следующее назначение.

Порт	Операция	Описание								
3B4h	Запись	Регистр индекса ЭЛТ. Используется для выбора регистра ЭЛТ. Для записи регистра ЭЛТ Rn нужно выполнить команды: out 3B4h, n и out 3B5h, данные.								
3B5h	Запись/ чтение	Регистр данных ЭЛТ. Позволяет записывать все регистры ЭЛТ и читать регистры R12 - R17.								
3B8h	Запись	Регистр режимов дисплея:								
		<div style="display: flex; align-items: center; margin-bottom: 5px;"> 7 6 5 4 3 2 1 0 <div style="display: flex; align-items: center;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;">X</td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;">X</td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;">X</td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;">X</td> </tr> </table> Бит: </div> </div> <ul style="list-style-type: none"> <li style="margin-bottom: 5px;">> 1: 0=текстовый режим, 1=графический режим <li style="margin-bottom: 5px;">> 3: 0=очистить экран, 1=активизировать экран <li style="margin-bottom: 5px;">> 5: 1=бит 7 атрибутов - мерцание 0=бит 7 атрибутов - яркость фона <li style="margin-bottom: 5px;">> 7: номер текущей видеостраницы 		X		X		X		X
	X		X		X		X			
3B9h	Запись/ Чтение	Вывод любого байта включает триггер светового пера.								
3BAh	Чтение	Регистр состояния дисплея:								
		<div style="display: flex; align-items: center; margin-bottom: 5px;"> 7 6 5 4 3 2 1 0 <div style="display: flex; align-items: center;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;">X</td> <td style="width: 20px; height: 20px;">X</td> <td style="width: 20px; height: 20px;">X</td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;">X</td> <td style="width: 20px; height: 20px;">X</td> <td style="width: 20px; height: 20px;"></td> </tr> </table> Бит: </div> </div> <ul style="list-style-type: none"> <li style="margin-bottom: 5px;">> 0: 0=вывод изображения, 1=горизонтальная синхронизация <li style="margin-bottom: 5px;">> 3: 0=экран включен, 1=выключен (позволяет проверить поступление видеосигнала на экран) <li style="margin-bottom: 5px;">> 7: 0=вертикальная синхронизация, 1=дисплей активен 		X	X	X		X	X	
	X	X	X		X	X				
3BBh	Запись/ Чтение	Вывод любого байта выключает триггер светового пера.								
3BCh-3BEh		См. описание портов параллельного принтера (п. 10).								

3BFh Запись Регистр конфигурации. Сюда заносится новая конфигурация дисплея:
 0 (DIAG) - работа только в текстовом режиме;
 1 (HALF) - допустима работа только с графической страницей 0;
 3 (FULL) - доступны обе графические страницы.

Видеопамять MGA располагается в адресах ОЗУ В0000h - ВFFFFh. В текстовом режиме (25 строк по 80 символов в строке) она занимает 4К с В0000h по В0FFFh. При этом адрес кода символа равен В000h : 160*L+2*C, адрес атрибута на единицу больше. Здесь L - номер строки (0 - 24), C - номер столбца (0 - 79).

Графический режим имеет разрешение 720 точек по горизонтали на 348 точек по вертикали, поэтому один графический экран требует 32К видеопамяти. MGA имеет две графические страницы: страница 0 размещается в адресах В0000h - В7FFFh, страница 1 - в адресах В8000h - ВFFFFh. Смещение от начала страницы до байта, содержащего точку с координатами (X,Y), равно

$$2000h * (Y \% 4) + 90 * \text{int}(Y/4) + \text{int}(X/8),$$

а номер соответствующего бита в этом байте равен 7 - (X % 8). Здесь X - горизонтальная позиция точки (0 - 719), Y - ее вертикальная позиция (0 - 347).

В качестве примера приведем программу инициализации MGA в текстовом или графическом режиме.

```

; Таблицы значений регистров ЭЛТ для двух режимов
Graph_Tbl db 35h,2Dh,2Eh,07h,5Bh,02h,57h,57h,02h,03h,00h,00h
Text_Tbl db 61h,50h,52h,0Fh,19h,06h,19h,19h,02h,0Dh,0Bh,0Ch

```

```

; Установка графического режима:

```

```

Gr_Mode:
mov al,2
lea si,Graph_Tbl
mov bx,0
mov cx,4000h
call Set_Mode
. . .

```

```

; Установка текстового режима:

```

```

Txt_Mode:
mov al,20h
lea si,Text_Tbl
mov bx,0720h
mov cx,2000h
call Set_Mode
. . .

```

```

; Подпрограмма инициализации MGA.

```

```

; Вход: AL - режим
; SI - адрес таблицы регистров ЭЛТ
; CX - размер экрана в словах
; BX - значение для очистки экрана

```

```

Set_mode proc near
push ax ; сохранить регистры
push bx
push cx
mov dx,3B8h ; регистр режимов
out dx,al ; изменить режим адаптера, отключив экран
mov ax,cs

```



```

    mov ds,ax      ; DS:SI = адрес таблицы
    mov dx,3B4h   ; регистр индекса
    mov cx,12     ; 12 регистров ЭЛТ
    xor ah,ah
    cld
Set_CRT:
    mov al,ah
    out dx,al     ; номер регистра ЭЛТ
    inc dx
    lodsb
    out dx,al     ; содержимое регистра ЭЛТ
    dec dx
    loop Set_CRT
    pop cx
    mov ax,0B000h
    mov es,ax
    pop ax
    xor di,di     ; ES:DI => видеопамять
    rep stosw    ; очистка экрана
    mov dx,3B8h   ; регистр режимов
    pop ax
    add al,8
    out dx,al     ; активизировать экран
    ret
Set_Mode endp

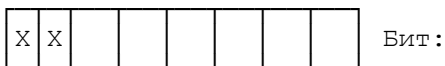
```

7.5. Цветной графический адаптер CGA

Цветной графический адаптер (CGA - Color Graphics Adapter) - это графический видеоконтроллер среднего разрешения. Для него зарезервированы порты ввода-вывода 3D0h - 3Dfh, имеющие следующее назначение.

Порт	Операция	Описание
3D4h	Запись	Регистр индекса ЭЛТ. Используется для выбора регистра ЭЛТ. Для записи регистра ЭЛТ Rn нужно выполнить команды: out 3D4h,n и out 3D5h,данные.
3D5h	Запись/ чтение	Регистр данных ЭЛТ. Позволяет записывать все регистры ЭЛТ и читать регистры R12 - R17.
3D8h	Запись	Регистр режимов дисплея:

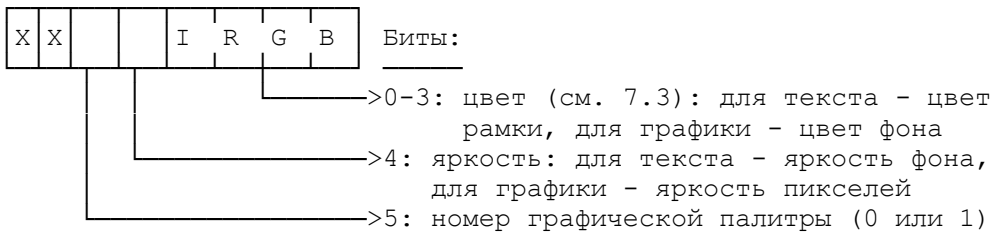
7 6 5 4 3 2 1 0



- >0: кол-во столбцов в текстовых режимах:
0=40 столбцов, 1=80 столбцов
- >1: 0=текстовый режим, 1=графический режим
- >2: 1=запрет цвета на композитном дисплее
- >3: 0=очистить экран, 1=активизировать экран
- >4: разрешение по горизонтали:
0=320 точек, 1=640 точек
- >5: 1=бит 7 атрибутов - мерцание
0=бит 7 атрибутов - яркость фона

3D9h Запись Регистр выбора цветов:

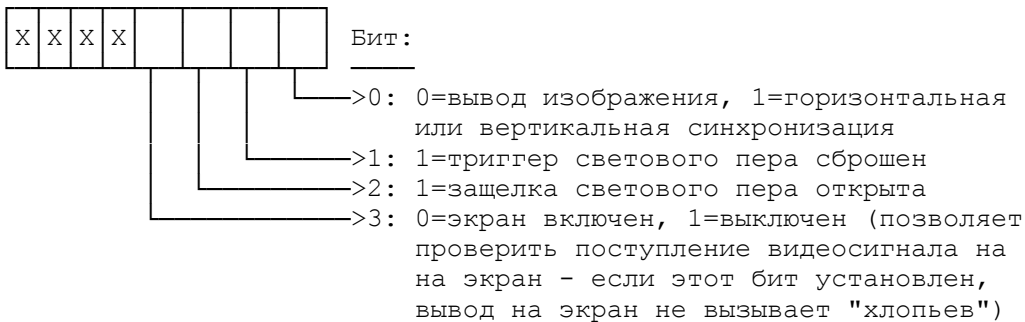
7 6 5 4 3 2 1 0



Примечание: Выбор палитры задает набор основных цветов в графическом режиме. Палитра 0 - зеленый/красный/коричневый, палитра 1 - бирюзовый/сиреневый/светло-серый (т. е. отличается добавлением синего цвета).

3DAh Чтение Регистр состояния дисплея:

7 6 5 4 3 2 1 0



3DBh Запись/ Чтение Вывод любого байта выключает триггер светового пера.

3DCh Запись/ Чтение Вывод любого байта включает триггер светового пера.

Видеопамять CGA располагается в адресах ОЗУ В8000h - ВFFFFh. CGA имеет два типа текстовых режимов: 25 строк по 40 символов в строке и по 80 символов в строке. В режимах 25 x 40 одна страница занимает 800h байтов, поэтому видеопамять состоит из 8 страниц; в режимах 25 x 80 страница занимает 1000h байтов и видеопамять состоит из 4 страниц.

Адрес кода символа в текстовых режимах равен В800h : $n * P + m * L + 2 * C$, адрес атрибута на единицу больше. Здесь L - номер строки (0 - 24), C - номер столбца (0 - 79 или 0 - 39), m = 2 * количество столбцов режима, n = 800h при m=80 и 1000h при m=160.

CGA имеет два типа графических режимов: цветной режим, имеющий разрешение 320 точек по горизонтали на 200 точек по вертикали, и черно-белый режим с разрешением 640 точек на 200 точек. В режиме 320 x 200 каждый пиксель задается двумя битами, кодирующими его цвет; в режиме 640 x 200 цвет пикселя задается одним битом (белый/черный). В графических режимах один экран требует 16К видеопамати, поэтому имеется одна страница видеопамати, которая размещается в адресах В8000h - ВFFFFh. Эта страница расслоена на две области: область с адреса В8000h содержит пиксели, имеющие четные номера строк; область с адреса ВА000h содержит пиксели из нечетных строк. Поэтому адрес байта, содержащего точку с координатами (X,Y), равен в цветном режиме

$$B800h : 2000h * (Y \% 2) + 80 * \text{int}(Y/2) + \text{int}(X/4),$$

а номера соответствующих битов в этом байте равны $7 - 2 * (X \% 4)$

4) и $6 - 2 * (X \% 4)$. Здесь X - горизонтальная позиция точки (0 - 319), Y - ее вертикальная позиция (0 - 199).

Адрес байта, содержащего точку с координатами (X,Y), равен в черно-белом режиме

$$B800h : 2000h * (Y \% 2) + 80 * \text{int}(Y/2) + \text{int}(X/8),$$

а номер соответствующего бита в этом байте равен $7 - (X \% 8)$. Здесь X - горизонтальная позиция точки (0 - 639), Y - ее вертикальная позиция (0 - 199).

7.6. Расширенный графический адаптер EGA

Расширенный графический адаптер (EGA - Enhanced Graphics Adapter) - это графический видеоконтроллер высокого разрешения, программно совместимый с адаптерами CGA и MGA, но имеющий гораздо большие возможности. В IBM PC для EGA зарезервированы следующие порты ввода-вывода:

EGA #1 - порты 3C0h - 3DFh;
EGA #2 - порты 2C0h - 2DFh.

EGA может генерировать прерывание IRQ2 (Int 0Ah) в начале вертикальной синхронизации. Однако, BIOS не использует эту возможность; мы также не описываем ее, т. к. вектор Int 0Ah используется в PC AT как каскад второго контроллера прерываний (см. п. 3).

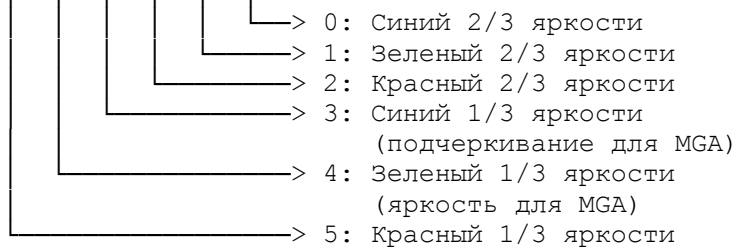
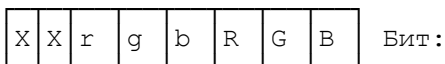
Все регистры EGA можно разбить на четыре группы:

3C0h	Контроллер атрибутов (ATC)
3C2h	Многоцелевой регистр
3C4h-3C5h	Блок синхронизации (sequencer)
3CAh-3CFh	Графический контроллер (GDC)
3D4h-3DCh или 3B4h-3BCh	Блок управления дисплеем (адреса портов зависят от видеорежима)

Контроллер атрибутов (ATC) имеет 20 регистров, управляющих цветами вывода:

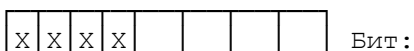
Регистр	Наименование	Описание
R0-R15	Регистры палитры (только для записи)	Определяют цвета соответствующих атрибутов:

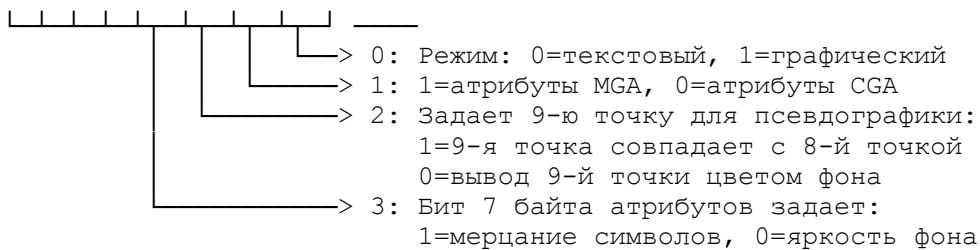
7 6 5 4 3 2 1 0



R16 Управление режимами Имеет следующую структуру:

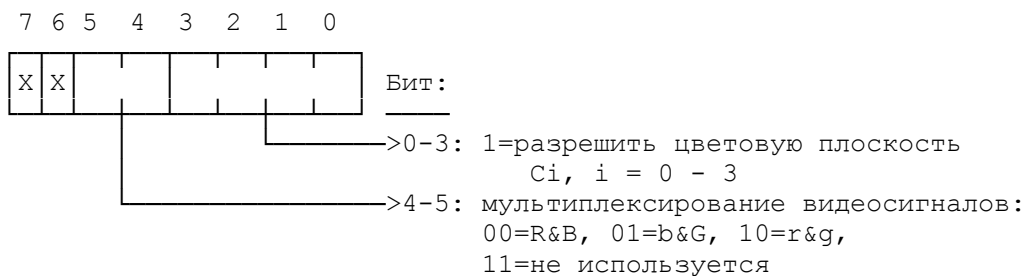
7 6 5 4 3 2 1 0





R17 Цвет рамки Устроен так же, как регистры палитры.

R18 Управление цветовыми плоскостями Имеет следующую структуру:



R19 Горизонтальная прокрутка Задаёт сдвиг экрана влево в точках (0 - 15).

Примечания.

1. Регистры палитры задают соответствие между значением атрибута и его цветом. Например, засылка значения 02h в регистр АТС определит его как зелёный цвет, 10h - как бледно-зелёный, 12h - как ярко-зелёный.

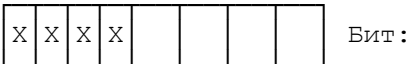
2. Для того, чтобы цвета атрибутов совпадали с цветами CGA, регистры палитры должны иметь следующие значения:

Регистр	Значение	Регистр	Значение
00h	00h	08h	38h
01h	01h	09h	39h
02h	02h	0Ah	3Ah
03h	03h	0Bh	3Bh
04h	04h	0Ch	3Ch
05h	05h	0Dh	3Dh
06h	14h	0Eh	3Eh
07h	07h	0Fh	3Fh

Блок синхронизации обеспечивает синхронизацию вывода символов с работой знакогенератора. Он имеет следующие программируемые регистры.

Регистр	Наименование	Описание
R0	Перезапуск блока синхронизации	Установка и сброс бита 0 вызывает асинхронный перезапуск блока, установка и сброс бита 1 - синхронный перезапуск. Для нормальной работы оба бита должны быть установлены.
R1	Режим синхронизации	Имеет следующую структуру:

7 6 5 4 3 2 1 0



Бит:

- > 0: Высота символов: 1=8 точек, 0=9 точек
- > 1: Полоса частот ЭЛТ: 1=низкая, 0=высокая
- > 2: 0=сдвиг всех символов, 1=только четных
- > 3: Частота генератора пикселей:
1=половинная, 0=обычная

R2 Маска битовых плоскостей Установка битов 0 - 3 разрешает запись в соответствующие битовые плоскости.

R3 Выбор карты символов Биты 0-1 выбирают карту В, биты 2-3 - карту А; карта А относится к символам, у которых бит 3 атрибутов установлен, карта В - к символам, чей бит 3 сброшен. Действительно только в случае, когда ОЗУ EGA больше 64К.

R4 Режим видеопамати Имеет следующую структуру:

7 6 5 4 3 2 1 0



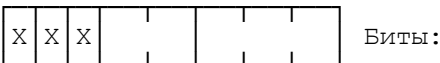
Бит:

- > 0: 1=знакогенератор текста, 0=графика
- > 1: Размер ОЗУ EGA: 0=64К, 1=более 64К
- > 2: Способ хранения пикселей:
1=последовательный, 0=как в CGA
(расслоение четных и нечетных строк)

Графический контроллер (GDC) управляет работой адаптера в графическом режиме и имеет следующие регистры:

Регистр	Наименование	Описание
R0	Установка/сброс битовых плоскостей	Установка битов 0 - 3 выбирает соответствующую битовую плоскость для режима записи 00.
R1	Разрешение установки и сброса	Управляет доступом к R0.
R2	Сравнение цветов	Биты 0 - 3 выбирают цвет для режима чтения 1.
R3	Параметры записи	Имеет следующую структуру:

7 6 5 4 3 2 1 0



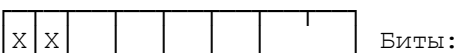
Биты:

- > 0-2: Счетчик циклических сдвигов влево для режима записи 00
- > 3-4: Способ вывода в режимах записи 00 и 02: 00=без изменений, 01=И, 10=ИЛИ, 11=исключающее ИЛИ

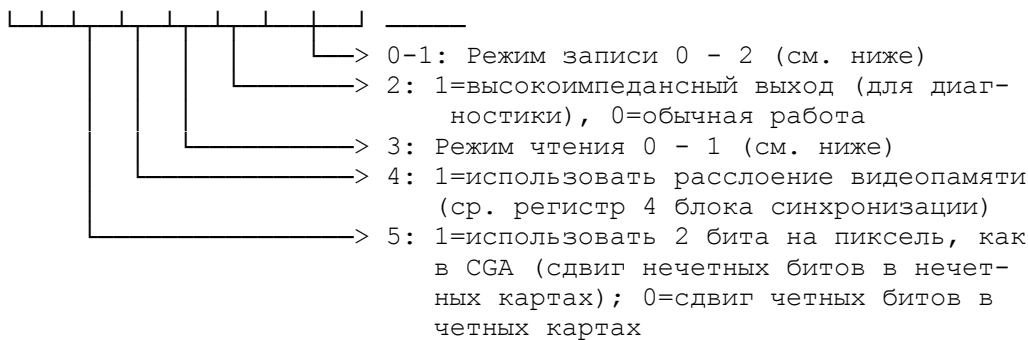
R4 Выбор карты чтения Биты 0-2 задают номер плоскости для режима чтения 1.

R5 Регистр режима Имеет следующую структуру:

7 6 5 4 3 2 1 0

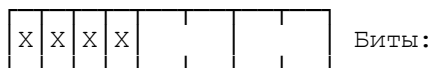


Биты:



R6 Управление графикой Имеет следующую структуру:

7 6 5 4 3 2 1 0



- > 0: 1=графика, 0=знакогенератор
- > 1: Ставить нечетные карты после четных
- > 2-3: Адрес и размер видеопамати:
00=A0000h (128K), 01=A0000h (64K)
10=B0000h (32K), 11=B8000h (32K)

R7 Маскирование цветов Биты 0-3 исключают соответствующие битовые плоскости в режиме чтения 1.

R8 Битовая маска Биты 0-7 задают биты, маскируемые во всех битовых плоскостях.

Примечание. Режимы чтения и записи имеют следующий смысл.
Режимы чтения:

- 0 - использовать выбранную битовую плоскость;
- 1 - сравнивать цвет пикселя с цветом в регистре R2 и возвращать 1 при их совпадении.

Режимы записи:

- 00 - циклически сдвинуть данные, применить способ вывода, записать в разрешенные плоскости;
- 01 - вывести в каждую плоскость значение, полученное предыдущим чтением;
- 10 - выводить в каждую плоскость 8 бит значения соответствующих битов 0-3 данных (быстрая закраска)
- 11 - недопустим.

Порты ввода-вывода EGA имеют следующее назначение.

Порт	Операция	Описание
------	----------	----------

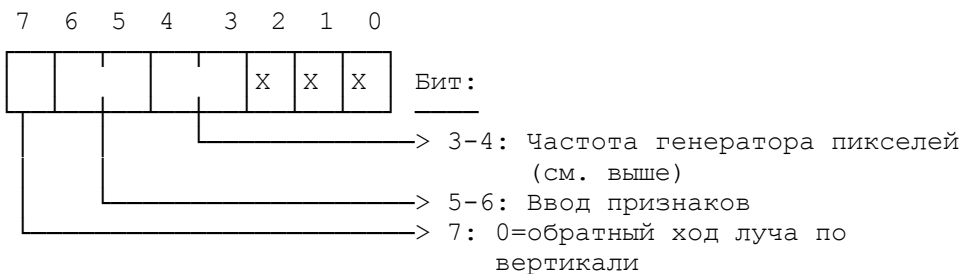
3C0h	Запись	Регистр контроллера атрибутов. Для доступа к регистрам АТС подайте команды: in al,3C2h ; перейти в адресный режим out 3C0h,n ; номер регистра АТС (режим данных) out 3C0h,val ; новое значение регистра Биты 0-4 в n - номер регистра, бит 5 задает: 1=разрешить вывод на экран, 0=установка регистра АТС.
------	--------	---

3C2h Запись Многоцелевой управляющий регистр:

7 6 5 4 3 2 1 0



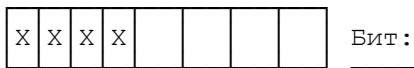
Чтение Регистр состояния 0:



3C4h	Запись	Регистр адреса блока синхронизации.
3C5h	Запись	Регистр данных блока синхронизации. Для программирования регистра Rn блока нужно выполнить команды: out 3C4h,n и out 3C5h,данные
3CAh	Запись	Графическая позиция 2 (для EGA должна быть 1).
3CCh	Запись	Графическая позиция 1 (для EGA должна быть 0).
3CEh	Запись	Регистр адреса графического контроллера.
3CFh	Запись	Регистр данных графического контроллера. Для программирования регистра Rn нужно выполнить команды: out 3CEh,n и out 3CFh,данные
3D4h или 3B4h	Запись	Регистр индекса ЭЛТ. Используется для выбора регистра ЭЛТ. Для записи регистра ЭЛТ Rn нужно выполнить команды: out 3X4h,n и out 3X5h,данные.
3D5h/ 3B5h	Запись/ чтение	Регистр данных ЭЛТ. Позволяет записывать и читать все регистры ЭЛТ.
3DAh/ 3BAh	Запись	Биты 0 - 1 управляют выходами признаков FC0 и FC1 (контакты 19 и 17 соответственно).

Чтение Регистр состояния 1:

7 6 5 4 3 2 1 0



Бит:

- >0: 0=вывод изображения, 1=горизонтальная или вертикальная синхронизация
- >1: 1=триггер светового пера сброшен
- >2: 1=защелка светового пера открыта
- >3: 0=экран включен, 1=выключен (EGA не требует проверки этого бита при выводе - в любом случае "хлопьев" не будет)

3DBh/ Запись/ Вывод любого байта выключает триггер светового
3BBh Чтение пера.

3DCh/ Запись/ Вывод любого байта включает триггер светового
3BCh Чтение пера.

EGA работает либо в режиме совместимости с CGA или MGA, либо в одном из своих собственных графических видеорежимов:

- 320 x 200, 16 цветов
- 640 x 200, 16 цветов
- 640 x 350, черно-белый
- 640 x 350, 16 цветов

В этих режимах видеопамять разбивается на 4 битовые плоскости, каждая из которых содержит образ экрана для одного из основных цветов: красного, зеленого, синего и яркости. Контроллер EGA считывает биты всех четырех плоскостей, соответствующие данному пикселю, и формирует его результирующий цвет на экране (комбинация из 4 двоичных цифр дает 16 вариантов цветов).

Видеопамять EGA отображается в адреса ОЗУ A0000h - B0000h. Размер памяти EGA варьируется от 64К до 256К в зависимости от комплектации адаптера. Однако, даже если память EGA равна 256К, она размещается в 16К адресного пространства оперативной памяти. Это объясняется тем, что каждая битовая плоскость отображается в одни и те же адреса ОЗУ; для выбора битовой плоскости, с которой Вы хотите обмениваться данными, необходимо переустанавливать соответствующие регистры графического контроллера.

В графическом режиме 640 x 200 64К оперативной памяти разбивается на 4 страницы по 16К каждая. Адреса этих страниц равны соответственно A0000h, A4000h, A8000h и AC000h. Смещение от начала страницы до байта, содержащего пиксель с координатами (X,Y), равно $80*Y + \text{int}(X/8)$, а номер соответствующего бита в этом байте равен $7 - (X \% 8)$. Здесь X - горизонтальная позиция точки (0 - 639), Y - ее вертикальная позиция (0 - 199).

Графический режим 320 x 200 предоставляет пользователю 8 страниц размером по 8К. Смещение от начала страницы до байта, содержащего пиксель с координатами (X,Y), равно $40*Y + \text{int}(X/8)$, а номер соответствующего бита в этом байте равен $7 - (X \% 8)$. Здесь X - горизонтальная позиция точки (0 - 639), Y - ее вертикальная позиция (0 - 199).

В графических режимах 640 x 350 имеются две страницы с адресами A0000h и A8000h соответственно. В остальном этот режим аналогичен предыдущим.

В качестве примера приведем три подпрограммы работы с дисплеем EGA в режиме 640 x 200. Если Вы хотите использовать их в других режимах, достаточно поменять значение константы

PageSize на соответствующее значение.

PageSize equ 4000h ; размер страницы для режима 640 x 200

```
;
; Подпрограмма очистки экрана.
; Вход: CX = номер очищаемой страницы (0-3)
;
```

```
EGA_Cls proc near
    push ax
    push bx
    push cx
    push dx
    push es
    push di
    xor ax,ax ; вычисляем смещение до страницы
    mov bx,PageSize
    jcxz Page_0
High_Page:
    add ax,bx
    loop High_Page
Page_0:
    mov di,ax ; DI = адрес страницы в видеопамяти
    mov ax,0A000h
    mov es,ax ; ES:DI = полный адрес страницы
    mov ax,0FF08h
    mov dx,3CEh ; графический контроллер
    out dx,ax ; установить все битовые маски
    mov ax,0FF02h
    mov dx,3C4h ; блок синхронизации
    out dx,ax ; разрешить запись во все плоскости
    mov ax,3
    mov dx,3CEh ; графический контроллер
    out dx,ax ; задать нормальный режим работы
    mov cx,PageSize
    xor al,al
    rep stosb ; обнулить страницу
    pop di
    pop es
    pop dx
    pop cx
    pop bx
    pop ax
    ret
EGA_Cls endp
```

```
;
; Подпрограмма вывода пикселя на экран.
; Вход: AX = цвет пикселя
; BX = номер страницы
; CX = номер столбца (X)
; DX = номер строки (Y)
;
```

```
Put_Pixel proc near
    push ax
    push bx
    push cx
    push dx
    push es

    push ax
    push cx
```

```

push dx
mov ax,dx
mov cx,80
mul cx ; AX = 80 * Y
pop dx
pop cx
push cx
shr cx,1
shr cx,1
shr cx,1 ; CX = X / 8
add ax,cx ; AX = 80 * Y + X / 8
mov cx,bx ; номер страницы
mov bx,PageSize
jcxz Page_Done
Next_Page:
add ax,bx ; учтем смещение до страницы
loop Next_Page
Page_Done:
pop cx
mov bx,ax ; BX = адрес байта, содержащего пиксель
mov ax,0A000h
mov es,ax ; ES:BX = полный адрес байта
and cl,7 ; CL = X % 7
mov ah,80h
shr ah,cl ; AH = позиция пикселя в байте
mov dx,3CEh
mov al,8
out dx,ax ; разрешить запись только в этот бит
pop ax
mov ch,al ; цвет пикселя
mov dx,3C4h
mov ax,0FF02h
out dx,ax ; разрешить запись во все плоскости
mov al,es:[bx] ; читаем байт
xor al,al
mov es:[bx],al ; обнуляем пиксель
mov al,2
mov ah,ch ; цвет
and ah,0Fh ; приведем к диапазону 0-15
out dx,ax ; разрешить запись в соотв. плоскости
mov al,es:[bx] ; читаем байт
mov al,0FFh
mov es:[bx],al ; устанавливаем пиксель
mov ax,0FF02h
out dx,ax ; разрешить запись во все плоскости
mov ax,3
mov dx,3CEh ; графический контроллер
out dx,ax ; задать нормальный режим работы
mov ax,0FF08h
out dx,ax ; включить все биты
pop es
pop dx
pop cx
pop bx
pop ax
ret
Put_Pixel endp

```

```

;
; Подпрограмма чтения цвета пикселя.
; Вход: CX = номер столбца (X)
; DX = номер строки (Y)
; Выход: AX = цвет пикселя
;

```

```

Get_Pixel proc near
    push ax
    push bx
    push cx
    push dx
    push es

    push dx
    mov ax,205h
    mov dx,3CEh
    out dx,ax ; режимы: записи=01, чтения=0
    pop dx
    push cx
    push dx
    mov ax,dx
    mov cx,80
    mul cx ; AX = 80 * Y
    pop dx
    pop cx
    push cx
    shr cx,1
    shr cx,1
    shr cx,1 ; CX = X / 8
    add ax,cx ; AX = 80 * Y + X / 8
    mov cx,bx ; номер страницы
    mov bx,PageSize
    jcxz My_Page
Higher_Page:
    add ax,bx ; учтем смещение до страницы
    loop Higher_Page
My_Page:
    mov di,ax ; DI = адрес байта, содержащего пиксель
    mov ax,0A000h
    mov es,ax ; ES:DI = полный адрес байта
    pop cx
    and cl,7 ; CL = X % 7
    mov ch,80h
    shr ch,cl ; CH = позиция пикселя в байте
    mov bl,7
    sub bl,cl
    mov cl,bl ; CL = номер бита в байте
    xor bl,bl ; накопитель цвета
    mov ah,3 ; счетчик битовых плоскостей
    mov al,4 ; R4 = регистр выбора плоскости
    mov dx,3CEh ; графический контроллер
Read_Plane:
    out dx,ax ; выбираем плоскость AH
    mov bh,es:[di] ; читать байт
    and bh,ch ; выбрать бит
    shr bh,cl ; сдвинуть цвет в бит 0
    shl bl,1 ; сдвинуть выбранные цвета
    or bl,bh ; и добавить считанный
    dec ah ; следующая плоскость
    jge Read_Plane ; повторить для всех плоскостей
    mov al,bl
    xor ah,ah ; AX = цвет пикселя
    pop es
    pop dx
    pop cx
    pop bx
    pop ax
    ret
Get_Pixel endp

```

7.7. Графический видеомассив VGA

Адаптер VGA (Video Graphics Array) представляет собой дальнейшее развитие EGA. В основном он полностью совместим с адаптером EGA, поэтому мы приведем только его основные отличия.

1. Если EGA-дисплеи имеют цифровой вход, то VGA - это дисплей с аналоговым входом. Поэтому при формировании цвета пикселя из битовых плоскостей цвет складывается из трех основных цветов (красного, зеленого и синего), интенсивность каждого из которых может плавно варьироваться от нуля до максимального значения.

2. VGA имеет большее разрешение, чем EGA, и предоставляет пользователю три дополнительных режима:

640 x 480, 2 цвета
640 x 480, 16 цветов
320 x 200, 256 цветов

В режимах 640 x 480 имеется одна графическая страница размером 153600 байт. В режиме 320 x 200 с 256 цветами используется несколько иной метод формирования цвета пикселя, чем в других режимах; за счет этого видеопамять разбивается на 4 страницы размером по 16К. Этот режим возможен только на аналоговых дисплеях или дисплеях с переменной частотой.

3. Частота генератора пикселей VGA принимает значения 25 или 28 Мгц (в отличие от 14 и 16 Мгц у EGA).

4. VGA имеет дополнительный порт ввода 3CCh, позволяющий считать содержимое многоцелевого регистра (см. описание порта 3C2h в п. 7.6). Графические позиции 3CAh и 3CCh, имеющиеся в EGA, у VGA отсутствуют.

8. Контроллер накопителей на гибких дисках

8.1. Основные сведения о контроллере

Управление гибкими дисками в PC-подобных ПЭВМ осуществляется микросхемой 8272A фирмы Intel, выполняющей функции контроллера гибких дисков (Floppy Disk Controller, FDC). Информация на дискете хранится на дорожках, дорожки разбиты на сектора. В зависимости от типа дисководов число дорожек на одной стороне дискеты может быть равно 40 или 80, число секторов на дорожке колеблется от 8 до 26. Кроме того существуют дисководы, имеющие одну головку чтения/записи (при этом вторая сторона дискеты не используется). Контроллер управляет дисководами для дискет диаметром 8, 5.25 и 3.5 дюйма, поддерживает запись на дискету методами ЧМ и МЧМ, программируемую длину секторов (128, 256, 512 или 1024 байта), мультисекторные и мультидорожечные операции обмена с дисками, операции поиска. Одновременно к одному контроллеру может быть подключено до четырех дисководов любого из поддерживаемых типов. Обмен с диском может осуществляться как с использованием ПДП, так и без него. В последнем случае используются 2 метода: опрос состояния контроллера через регистры или использование аппаратных прерываний контроллера. FDC генерирует аппаратные прерывания уровня 6 (IRQ6), связанные с вектором прерываний 0Eh. Управление контроллером и накопителями осуществляется через 3 порта ввода-вывода.

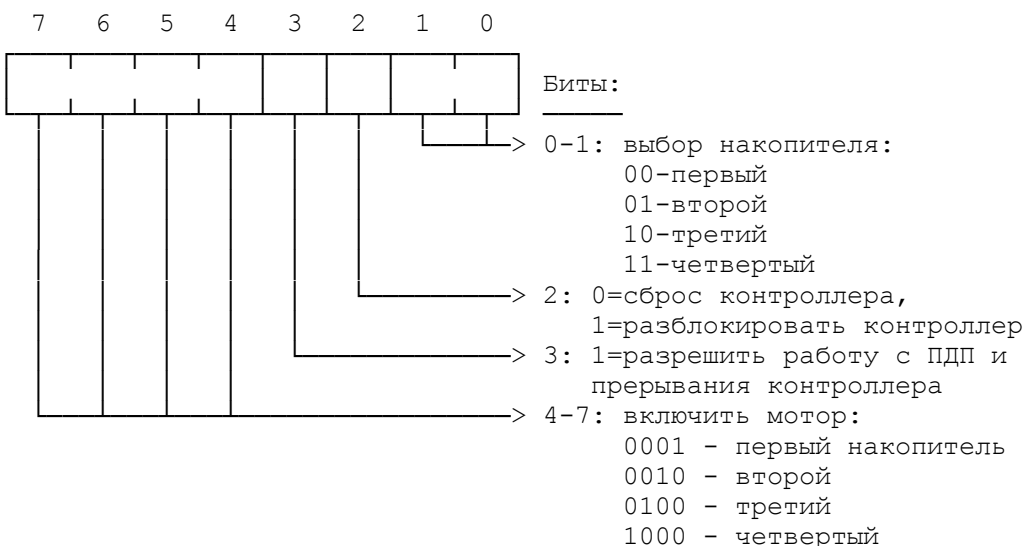
8.2. Порты контроллера накопителей на гибких дисках

Материалы данного и следующего разделов полностью описывают возможности контроллера 8272А, система команд которого является базовой для всех более поздних моделей. Дополнительные возможности контроллера 82072 не описаны, т. к. этот контроллер полностью поддерживает описываемую систему команд, а его неописанные функции практически не используются в стандартных средствах.

Назначение портов ввода-вывода контроллера гибких дисков приведено в таблице.

Порт	Режим	Назначение
3F2h	запись	Передача команд по управлению режимом работы контроллера и накопителей
3F4h	чтение	Чтение регистра состояния контроллера
3F5h	запись	Передача команд по управлению накопителями
3F5h	чтение	Чтение данных с диска в режиме работы без ПДП
3F5h	запись	Запись данных на диск в режиме работы без ПДП
3F5h	чтение	Чтение результатов выполнения команды во всех режимах

Порт 3F2h.

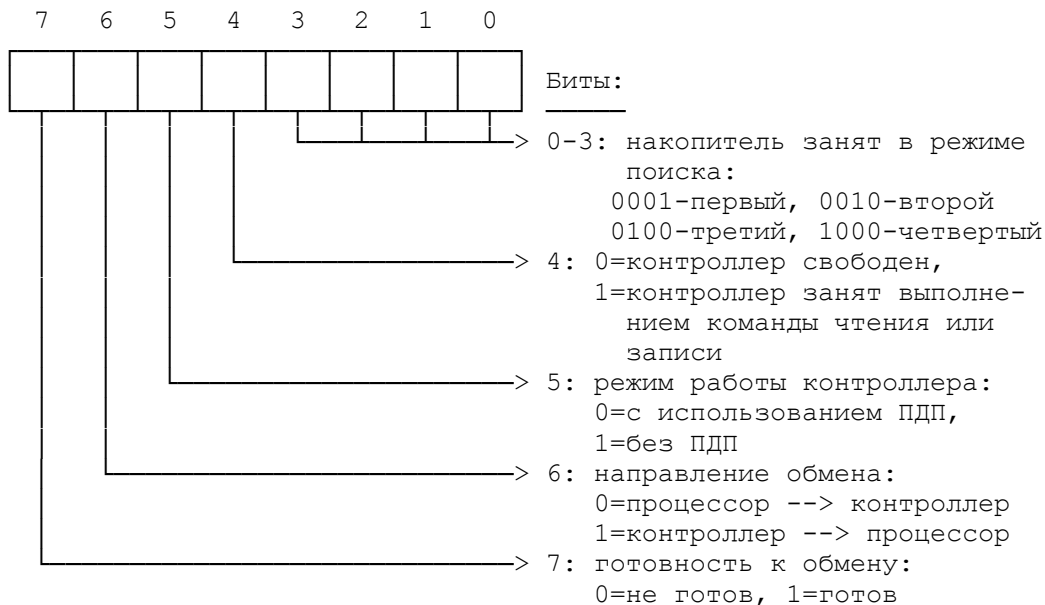


На АТ-подобных ПЭВМ обычно допускается не 4, а только 2 накопителя, вследствие чего комбинации, использующие третий и четвертый накопители, игнорируются. Бит 2, равный 0, прекращает выполнение текущей команды в момент вывода в порт. Если команда не завершилась, ее выполнение прерывается. Если не были считаны результаты выполнения команды, они пропадают. Если не закончился ввод команды, он пропадает и должен быть начат заново. В любом

случае продолжение работы возможно только после установки этого бита обратно в 1.

Порт 3F4h.

Через этот порт осуществляется чтение регистра основного состояния FDC.



Бит 7, равный 1, говорит о том, что контроллер при вводе команды готов к приему очередного байта команды, при обмене данными в режиме без ПДП готов к приему/передаче очередного байта данных, при приеме результата выполнения команды готов к передаче очередного байта результата.

8.3. Команды управления накопителями

Выполнение каждой команды имеет 3 фазы:

- фаза ввода команды;
- фаза выполнения;
- фаза результата.

В фазе ввода команды контроллер побайтно принимает и накапливает команду. Фаза команды заканчивается запуском мотора на соответствующем накопителе (это не относится к командам задания режимов и чтения состояния).

В фазе выполнения осуществляется обмен данными между ОЗУ и накопителем (если команда предполагает такой обмен). В режиме с использованием ПДП перед приемом/передачей байта данных генерируется сигнал "запрос на ПДП". Контроллер ПДП отвечает на него сигналом подтверждения и командой приема/передачи. В режиме работы без ПДП при готовности к приему/передаче каждого байта генерируется прерывание. Если прерывания запрещены, приграмма может в цикле проверять регистр состояния (порт 3F4h) до получения подтверждения готовности контроллера к обмену данными.

По окончании выполнения команды генерируется прерывание, означающее начало фазы результата. В фазе результата контроллер передает результаты выполнения команды. Все байты результата должны быть считаны, либо необходимо выполнить сброс контроллера, иначе следующая команда не будет принята. Не все

команды имеют фазу результата, ряд команд выполняется в 2 фазы.

8.3.1. Перечень команд контроллера НГМД

Ниже приводится подробное описание команд FDC. Фаза выполнения помечена знаками "*" * *". Смысл символических обозначений элементов команд приведен после описания команд. В битовых масках биты нумеруются справа налево. В фазе результата все байты результата приведены в той последовательности, в которой они возвращаются контроллером.

Команда	Регистр данных	Назначение
1. Чтение данных	MT MF SK 0 0 1 1 0 X X X X X HD US1 US0 C H R N EOT GPL DTL * * * ST0 ST1 ST2 C H R N	Ввод данных из поля данных сектора
2. Чтение удаленных данных	MT MF SK 0 1 1 0 0 X X X X X HD US1 US0 C H R N EOT GPL DTL * * * ST0 ST1 ST2 C H R N	Ввод данных из поля данных, помеченного как удаленное
3. Запись данных	MT MF 0 0 0 1 0 1 X X X X X HD US1 US0 C H R N EOT GPL DTL * * * ST0 ST1 ST2	Вывод данных в поле данных сектора

	SC GPL D * * *	информации
	ST0 ST1 ST2 C H R N	

Перед выполнением команды форматирования необходимо спозиционировать головку на требуемую дорожку. Данные для записи в идентификатор адреса сектора (цилиндр, головка, дорожка и размер сектора) принимаются от ПДП или через порт данных и не проверяются. Физический размер сектора определяется параметром "N" - код длины сектора, задаваемым в команде и одинаковым для всей дорожки. Таким образом нельзя иметь на одной дорожке сектора разной длины. Команда не заносит в маркер данных признак "удаленные данные", а заполняет поле данных секторов дорожки байтом-заполнителем.

8.Сканирование до "равно"	MT MF SK 1 0 0 0 1 X X X X X HD US1 US0 C H R N EOT GPL STP * * * ST0 ST1 ST2 C H R N	Сканирование данных на указанное условие и формирование запроса на прерывание при выполнении условия
9.Сканирование до "больше или равно"	MT MF SK 1 1 0 0 1 X X X X X HD US1 US0 C H R N EOT GPL STP * * * ST0 ST1 ST2 C H R	аналогично предыдущей команде
10.Сканирование до "меньше или равно"	MT MF SK 1 1 1 0 1 X X X X X HD US1 US0 C H R	аналогично предыдущей команде

	<pre> N EOT STP * * ST0 ST1 ST2 H R N * * * </pre>	нулевую дорожку
12. Определение	<pre> 0 0 0 0 0 0 1 1 <--- SRT ---> <--- HUT ---> <----- HLT -----> ND </pre>	Задание режима работы дисководов
13. Поиск	<pre> 0 0 0 0 1 1 1 1 X X X X X HD US1 US0 NCN * * * </pre>	Позиционирование головки на заданную дорожку
14. Чтение состояния прерывания	<pre> 0 0 0 0 1 0 0 0 * * * ST0 PCN </pre>	Чтение байта состояния ST0 после прерываний, вызванных изменением готовности дисководов и операцией поиска
15. Чтение состояния НГМД	<pre> 0 0 0 0 0 1 0 0 X X X X X HD US1 US0 * * * ST3 </pre>	Чтение байта состояния ST3
16. Несуществующая команда	<pre> < ошибочный код команды FDC > * * * ST0 </pre>	При ошибочной команде возвращается ST0=80h

Смысловое значение элементов команд:

MT=1 - многодорожечная (двусторонняя операция);

MF=1 - МЧМ-кодирование (двойная плотность) - пропуск синхроимпульсов у единичных битов и пропуск синхроимпульсов у нулевых битов, стоящих после единичных;

SK=1 - пропуск адресного маркера удаленных данных;

HD - номер головки (0 или 1);

US1, US0 - номер накопителя (0 - 3);

C - номер цилиндра;

H - номер головки;

R - номер сектора;

N - код длины сектора (0 - 128, 1 - 256, 2 - 512, 3 - 1024 байтов);

EOT - номер последнего сектора на цилиндре;

GPL - длина межзонного промежутка (обычно 2Ah при обмене и 50h при форматировании);

DTL - число считываемых/записываемых байт (действительно при N = 0);

ST0 - ST3 - байты состояния ST0-ST3;

STP - межсекторное смещение;

SRT - время скорости шага в миллисекундах (обычно 0Dh);

HUT - время разгрузки головки в единицах по 16 миллисекунд (обычно = 0Fh);

HLT - время загрузки головки в единицах по 2 миллисекунды (обычно = 1);

ND=0 - режим работы с использованием ПДП, =1 - режим работы без использования ПДП;

NCN - номер цилиндра после поиска;

PCN - номер текущего цилиндра;

SC - количество секторов на цилиндре;

D - символ-заполнитель для разметки (обычно 0F6h).

8.3.2. Структура байтов состояния

Байт состояния ST0:

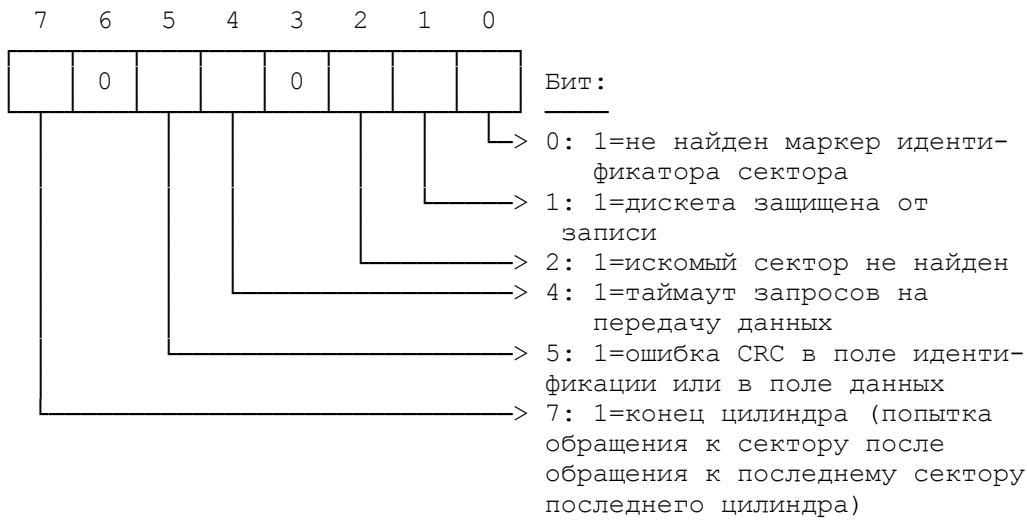


Бит 2 (номер головки) после выполнения команды поиска всегда равен 0, независимо от того, на какой головке осуществлялся поиск.

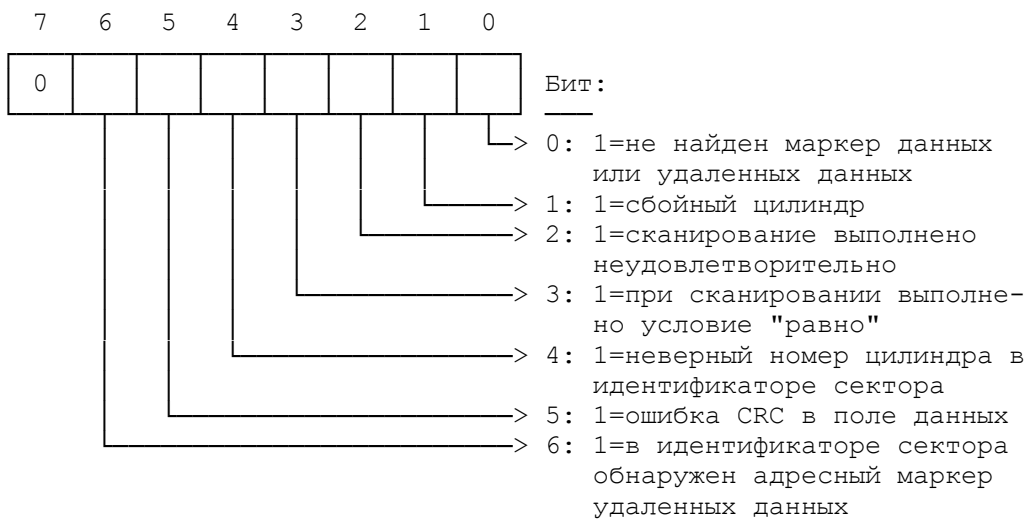
Бит 3 будет всегда равен 1, если на дисковом с одной

головой вы попытаете обратиться к стороне 1 (допустима только сторона 0).

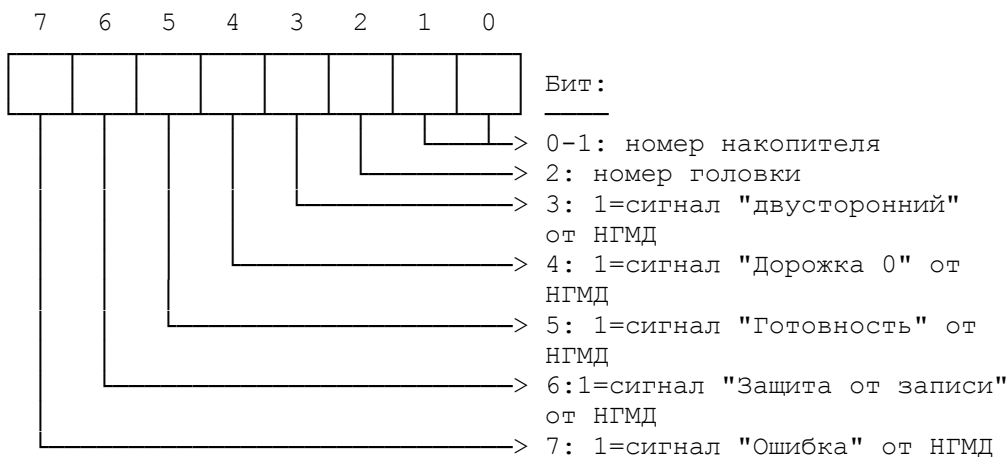
Байт состояния ST1:



Байт состояния ST2:



Байт состояния ST3:

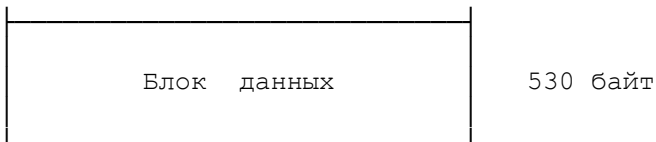


Индикатором начала дорожки служит индексное отверстие. Запись на дорожку осуществляется в несколько приемов. При форматировании дорожка перезаписывается полностью. При операциях записи перезаписываются отдельные сектора, что влечет за собой сбой намагниченности диска в месте обрыва записи, причем физическая длина сектора при его перезаписи не всегда точно совпадает с длиной ранее имевшегося на этом месте сектора, разница в длине может составлять до 4 - 5 бит даже при перезаписи на одном дисководе в одно время. Для синхронизации после сбоя на дорожке имеются зазоры и синхронизоны. В общем виде формат дорожки можно изобразить следующим образом:

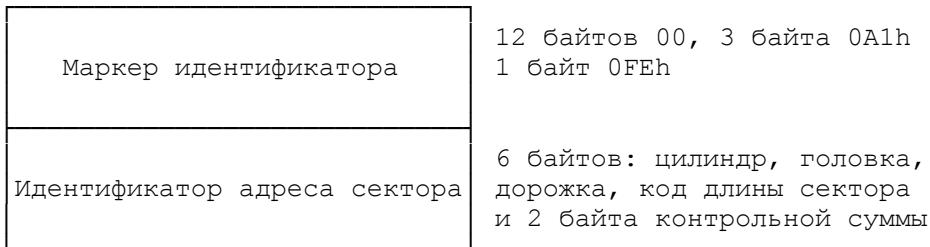
Индекс	Определяется индексным отверстием
Зазор индекса	80 байт 4Eh
Заголовок дорожки	12 байтов 00h, 3 байта 0C2h, 1 байт 0FCh, 50 байтов 4Eh
1-й сектор	574 байта
Межсекторный промежуток	80 байт 4Eh
. 8 или 9 секторов на дорожке. .	
Последний сектор	574 байта
Межсекторный промежуток	80 байт 4Eh
Зазор дорожки	байты 4Eh до индексного отверстия
Индекс	

Формат сектора:

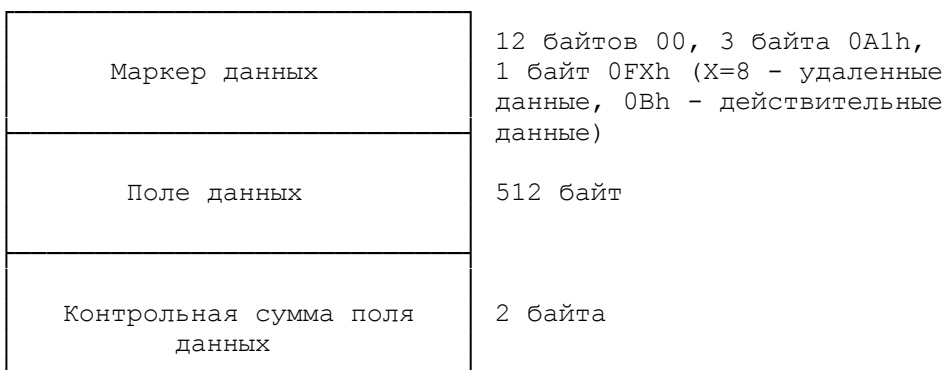
Идентификатор сектора	22 байта
Зазор идентификатора	22 байта 4Eh



Идентификатор сектора:



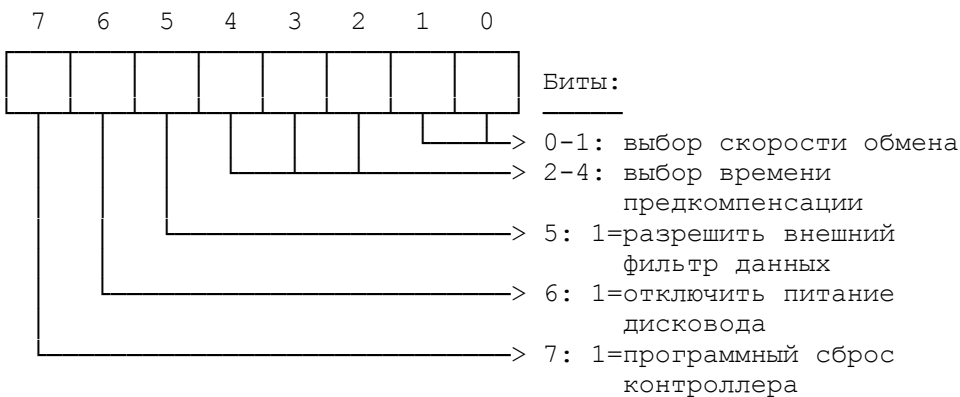
Блок данных:



8.5. Отличия контроллера для AT-подобных ПЭВМ

Архитектура AT-подобных ПЭВМ отличается от XT тем, что контроллер гибких дисков находится на одной плате с контроллером жестких дисков. К дополнительным возможностям относится возможность регулирования скорости обмена данными и возможность подключения двух контроллеров к одному ПЭВМ (в этом случае вместо портов 3F2h - 3F7h используются порты 372h - 377h). Функции контроллера гибких дисков на AT-подобных ПЭВМ выполняются микросхемой 82072 фирмы Intel (советских аналогов нет). Дополнительно к описанным выше портам имеется еще один порт 3F7h, позволяющий осуществить доступ к регистру выбора скорости обмена данными (Data Rate Select Register, DSR).

Порт 3F7h.



Возможные скорости обмена данными в зависимости от битов 0

- 1 DSR приведены в таблице.

Биты		Скорость	
1	0	МЧМ	ЧМ
1	1	1 Мбит/сек	Невозможно
0	0	500 Кбит/сек	250 Кбит/сек
0	1	300 Кбит/сек	150 Кбит/сек
1	0	250 Кбит/сек	125 Кбит/сек

В зависимости от значений битов 2, 3 и 4 DSR возможны следующие значения времени предкомпенсации:

Биты			Время предкомпенсации
4	3	2	
1	1	1	Недопустимая комбинация
0	0	1	41.67 нс
0	1	0	83.34 нс
0	1	1	125.00 нс
1	0	0	166.00 нс
1	0	1	208.33 нс
1	1	0	250.00 нс
0	0	0	Взять значение по умолчанию

Значения по умолчанию для времени предкомпенсации в зависимости от скорости приведены в таблице.

Скорость обмена	Время предкомпенсации
1 Мбит/сек	41.67 нс
500 Кбит/сек	125.00 нс
300 Кбит/сек	125.00 нс
250 Кбит/сек	125.00 нс

Кроме пятнадцати описанных выше команд контроллера 8272, 82072 поддерживает еще 4 команды:

- изменить параметры конфигурации;
- включение/выключение мотора;
- относительный поиск;
- вернуть состояние регистров контроллера.

Описание команд приведено в таблице.

Команда	Шина данных	Назначение

1. Изменить параметры конфигурации	<pre> 0 0 0 1 0 0 1 1 HDSA <- MOFF-> <--- MON ---> 0 EIS EF POLL <- FIFOTHR -> <----- PRETRK -----> * * * </pre>	Команда изменяет указанные параметры
2. Включение/выключение мотора	<pre> ON/OFF US1 US0 0 1 0 1 1 </pre>	Включает/выключает мотор на заданном дисковом
3. Относительный поиск	<pre> 1 DIR 0 0 1 1 1 1 0 0 0 0 0 HD US1 US0 <----- RCN -----> * * * </pre>	Осуществляется сдвиг головок в заданном направлении на заданное число цилиндров
4. Вернуть состояние регистров	<pre> 0 0 0 0 1 1 1 0 * * * <---- PCN-устр-ва 0 -----> <---- PCN-устр-ва 1 -----> <---- PCN-устр-ва 2 -----> <---- PCN-устр-ва 3 -----> <--- SRT ---> <--- HUT ---> <----- HLT -----> ND <----- SC/EOT -----> HDSA <- MOFF -> <--- MON ---> <----- PRETRK -----> 0 EIS EF POLL <- FIFOTHR -> </pre>	Возвращает состояние внутренних регистров контроллера

В таблице приняты следующие сокращения:

US1, US0 - номер устройства;

DIR - направление изменения (0-увеличение, 1-уменьшение);

HD - номер головки чтения/записи;

RCN - относительный номер цилиндра;

PCN - текущий номер цилиндра;

SRT - время скорости шага;

HUT - время разгрузки головки;

HLT - время загрузки головки;

ND - режим без ПДП;

SC/EOT - число секторов на дорожке/номер последнего сектора на дорожке;

ON/OFF - включение/выключение мотора (0/1);

MON - задержка перед включением мотора;

MOFF - задержка перед выключением мотора;

EIS = 1 - разрешить автоматический поиск перед чтением/записью данных, по умолчанию = 0;

EF = 1 - переводит контроллер в режим имитации 8272A;

POLL = 1 - запрещает обращение к устройствам;

HDSA = 1 - позволяет изменять временные задержки в зависимости от скорости вращения дисководов;

PRETRK - номер дорожки для предкомпенсации (обычно = 0).

9. Контроллер накопителей на жестких дисках

В данном разделе описывается управление контроллером накопителей на жестких магнитных дисках типа "винчестер". Материалы раздела подобраны в основном из технической документации и программного обеспечения для жестких дисков фирмы Seagate, однако данных о том, что существуют значительные отличия в управлении дисками, выпускаемыми другими фирмами, в настоящий момент не имеется. В связи с тем, что управление жесткими дисками в XT и AT-подобных ПЭВМ принципиально отличается, описание XT и AT контроллеров приведены отдельно друг от друга.

Часть информации, полученная из анализа программного обеспечения, возможно является неполной. Такая информация помечена звездочкой (*) в конкретной строке изложения или в заголовке раздела.

9.1. Контроллер жестких дисков для XT-подобных ПЭВМ

Архитектура XT-подобных ПЭВМ допускает подключение к одному контроллеру до двух накопителей. Обмен с жесткими дисками производится с использованием третьего канала ПДП. Контроллер генерирует аппаратные прерывания 5-го уровня (IRQ5), вектор прерывания 0Dh. Работа с диском осуществляется с использованием отдельного ПЗУ, содержащего процедуры управления диском, что является характерной особенностью XT-подобных ПЭВМ. Обычно в качестве сегментного адреса ПЗУ используется 0C800h, но возможны и другие значения (0CA00h, 0D000h или 0D800h для дисков ST-251 или ST-225 фирмы Seagate). Имеется 16 адресов портов ввода-вывода, предназначенных для управления контроллером и накопителями (320h-32Fh), однако реально используются только 4 из них, например 320h-323h, 324h-327h, 328h-32Bh или 32Ch-32Fh. Назначение портов (1-й, 2-й, 3-й и 4-й в каждой группе) одно и то же, поэтому далее будут описаны только порты 320h-323h.

9.1.1. Порты ввода-вывода

Назначение портов контроллера жестких дисков приведено в таблице.

Порт	Режим	Назначение
320h	запись	Передача команд по управлению накопителями
320h	чтение	Чтение параметров состояния после ошибок, возникших в процессе выполнения последней команды
321h	запись	Сброс контроллера

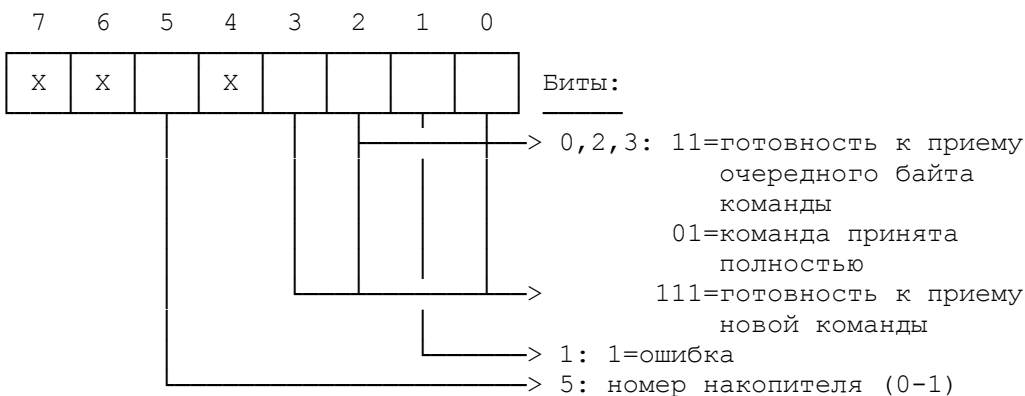
321h	чтение	Чтение состояния контроллера
322h	запись	Разблокировка (выбор) контроллера
322h	чтение	Чтение типа накопителя
323h	запись	Установка режима работы контроллера

Порт 320h.

Через этот порт контроллер принимает команды по управлению накопителями. Команда содержит до шести байтов информации, первый из которых определяет класс и код команды, а остальные содержат требуемые для выполнения команды параметры. При возникновении ошибки в процессе выполнения команды (бит 1 порта 321h) следует ввести в порт 320h команду "запрос состояния", после чего прочитать из него 4 байта состояния, первый из которых определяет код ошибки.

Порт 321h.

Запись нулевого байта в этот порт вызывает сброс контроллера. При чтении состояния контроллера возможны следующие значения битов:



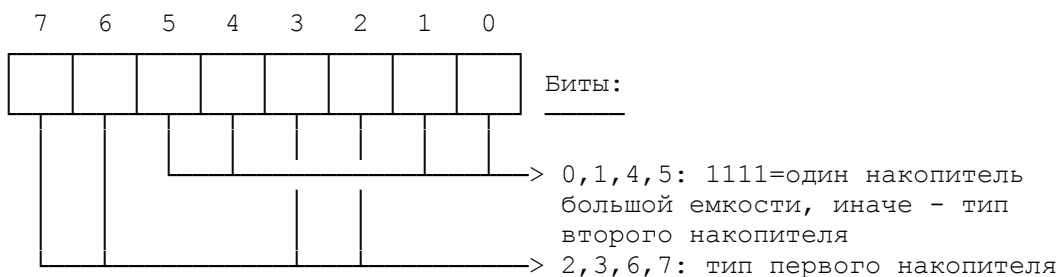
Бит 2, равный 1, вместе с битом 0, равным 1, указывает на то, что контроллер готов к приему очередного байта команды. Если в ходе передачи команды бит 2 = 0, а бит 0 = 1, значит контроллер считает, что он принял команду полностью и не ждет ввода новых байтов команды.

Бит 3, равный 1, вместе с битами 0 и 1, равными 1, указывает на готовность очередного байта состояния при выполнении команды "запрос состояния". Если установлены в 1 биты 0, 2 и 3 - это означает, что контроллер свободен и готов к приему первого байта новой команды.

Порт 322h.

Вывод произвольного байта в этот порт разблокирует контроллер. Операция разблокирования (выбора, активизации) контроллера (enable controller) должна выполняться каждый раз перед началом вывода через порт 320h новой команды.

При чтении из порта 322h может быть прочитан байт 0FFh, что не несет никакой информации, либо тип накопителя (применительно к данной версии ПЗУ).



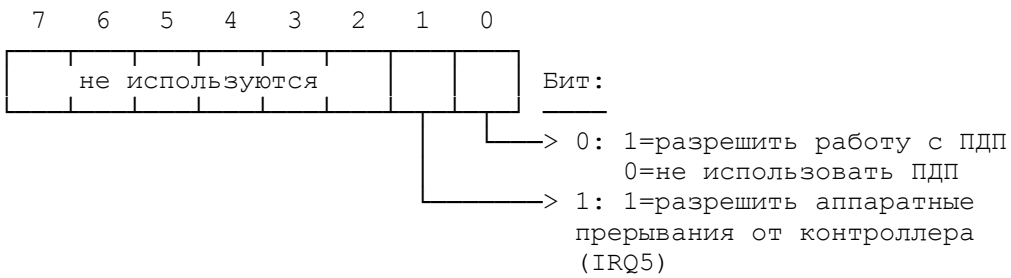
Если тип второго накопителя (биты 5, 4, 1 и 0) равен 1111, это означает, что к контроллеру подключен один накопитель большой емкости (более 32 Мбайт), который при форматировании следует разбивать на несколько логических дисков (говорят, что установлен ключ разбиения (Split switch)). Стандартное ПЗУ XT фирмы Seagate может работать со следующими 15-ю типами накопителей (номера 0 - 14), приведенными в таблице. Типом 15 (неопределенный тип) отмечаются все накопители, отличные по параметрам от описанных.

Тип	Фирма изготовитель и модель диска	C	H	R	B	V
0	Seagate ST-225	615	4	17	0	21.4
1	Seagate ST-4038	733	5	17	2	31.9
2	Miniscribe-3425	615	4	17	0	21.4
3	Seagate ST-212	306	4	17	0	10.6
4	Priam V-150	987	5	17	0	42.9
5	Priam V-170	987	7	17	0	60.1
6	Miniscribe-8425, Kyocera 20A/20B	615	4	17	0	21.4
7	Seagate ST-138	615	6	17	0	32.1
8	Seagate ST-4051	977	5	17	2	42.5
9	Seagate ST-251/ST-251-1	820	6	17	0	42.8
10	HH-725	612	4	17	0	21.3
11	Seagate ST-4053, Miniscribe-3053/6053	1024	5	17	0	44.5
12	Maxtor-1085, Newbury Data-1085 Miniscribe-6085	1024	8	17	0	71.3
13	Seagate ST-4096	1024	9	17	0	80.2
14	Maxtor-1140, Newbury Data-1140	396	15	17	0	51.7

Здесь

- C-число цилиндров (десятичное);
- H-число головок (десятичное);
- R-число секторов на дорожке (десятичное);
- B-стандартный управляющий байт (шестнадцатеричный);
- V-емкость диска (Мбайт).

Порт 323h.



Установка режима работы производится обычно до разблокировки контроллера перед началом вывода команды.

9.1.2. Общий формат команды и терминология

Команда контроллера может состоять из 1 - 6 байт. Общий формат команды приведен на рисунке.

\ биты \ байты\ \	7	6	5	4	3	2	1	0
0	Класс команды			Код операции				
1	0	0	У	Номер головки				
2	Цилиндр (старшие биты)		Номер сектора					
3	Цилиндр (младшие биты)							
4	Фактор чередования или число секторов (зависит от команды)							
5	Управляющий байт							

Здесь и далее:

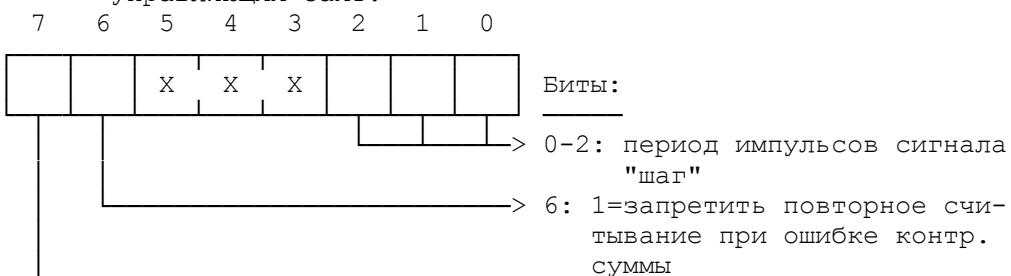
- класс команды и код операции определяют код команды контроллера;

- У - номер устройства (0/1);

- номер сектора - номер начального сектора для обмена;

- фактор чередования (Interleave Factor) задает последовательность чередования секторов на дорожке (см. формат дорожки);

- управляющий байт:



1	0	0	У	Х	Х	Х	Х	Х
2	Х	Х	Х	Х	Х	Х	Х	Х
3	Х	Х	Х	Х	Х	Х	Х	Х
4	Х	Х	Х	Х	Х	Х	Х	Х
5	П	0	0	0	0	Ш	Ш	Ш

--> Запрос состояния (класс 0, опкод 3).

Команда подается непосредственно после обнаружения ошибки (бит 1 порта 321h). Команда имеет длину 2 байта и вызывает передачу контроллером четырех байтов результата, определяющих параметры возникшей ошибки.

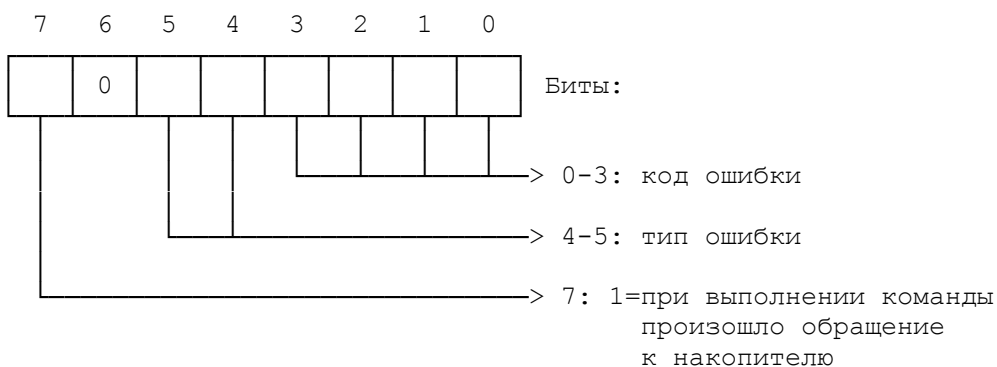
Формат команды:

0	0	0	0	0	0	0	1	1
1	0	0	У	Х	Х	Х	Х	Х

Формат результата:

0	байт состояния							
1	0	0	У	<----- НГ ----->				
2	ЦСВ		<----- НС ----->					
3	<----- ЦМБ ----->							

Формат байта состояния:



Бит 7 байта состояния равен 1, если при выполнении команды произошло обращение к накопителю, в этом случае байты 1 - 3 результата содержат корректную информацию, иначе эта информация некорректна.

Описание возможных ошибок приведено в таблице.

Тип ошибки	Код ошибки	Значение
------------	------------	----------

1	0	0	У	<-----	НГ	----->
2	ЦСВ	<-----	НС	----->		
3	<-----	ЦМБ	----->			
4	<-----	ЧС	----->			
5	П	ПС	0	0	0	Ш Ш Ш

--> Форматирование дорожки (класс 0, опкод 6).

Команда форматирует одну заданную дорожку. В остальном ее действие аналогично команде "форматирование накопителя".

0	0	0	0	0	0	1	1	0
1	0	0	У	<-----	НГ	----->		
2	ЦСВ	0	0	0	0	0	0	0
3	<-----	ЦМБ	----->					
4	0	0	0	<-----	ФЧ	----->		
5	П	0	0	0	0	Ш	Ш	Ш

--> Форматирование дефектной дорожки (класс 0, опкод 7).

Команда форматирует одну заданную дорожку, устанавливая флаги "дефектная дорожка" в заготовках секторов. Никакая информация в поля данных не записывается.

0	0	0	0	0	0	1	1	1
1	0	0	У	<-----	НГ	----->		
2	ЦСВ	0	0	0	0	0	0	0
3	<-----	ЦМБ	----->					
4	0	0	0	<-----	ФЧ	----->		
5	П	0	0	0	0	Ш	Ш	Ш

--> Чтение секторов (класс 0, опкод 8).

Команда считывает запрошенное число секторов начиная с заданного. Все сектора должны находиться на одной дорожке.

0	0	0	0	0	1	0	0	0
1	0	0	У	<-----	НГ	----->		
2	ЦСВ	<-----	НС	----->				
3	<-----	ЦМБ	----->					

4	<----- ЧС ----->							
5	П	ПС	0	0	0	Ш	Ш	Ш

--> Запись секторов (класс 0, опкод 0Ah). (*)

Команда записывает заданное в блоке параметров число секторов на диск, начиная с указанного начального сектора. Все записываемые сектора должны находиться на одной дорожке.

0	0	0	0	0	1	0	1	0
1	0	0	У	<----- НГ ----->				
2	ЦСВ		<----- НС ----->					
3	<----- ЦМБ ----->							
4	<----- ЧС ----->							
5	П	ПС	0	0	0	Ш	Ш	Ш

--> Поиск цилиндра по номеру (класс 0, опкод 0Ch). (*)

Команда осуществляет позиционирование головок накопителя на заданный цилиндр.

0	0	0	0	0	1	1	0	0
1	0	0	У	<----- НГ ----->				
2	ЦСВ	Х	Х	Х	Х	Х	Х	Х
3	<----- ЦМБ ----->							
4	Х	Х	Х	Х	Х	Х	Х	Х
5	П	0	0	0	0	Ш	Ш	Ш

--> Чтение внутреннего ОЗУ контроллера (класс 0, опкод 0Eh). (*)

Команда считывает содержимое внутреннего ОЗУ контроллера (512 байт). Данное ОЗУ используется контроллером при чтении/записи и содержит информацию, считанную/записанную при выполнении последней команды чтения/записи секторов.

0	0	0	0	0	1	1	1	0
1	0	0	Х	Х	Х	Х	Х	Х
2	Х	Х	Х	Х	Х	Х	Х	Х
3	Х	Х	Х	Х	Х	Х	Х	Х
4	Х	Х	Х	Х	Х	Х	Х	Х
5	Х	Х	Х	Х	Х	Х	Х	Х

--> Запись внутреннего ОЗУ контроллера (класс 0, опкод 0Fh). (*)

Команда обновляет содержимое внутреннего ОЗУ контроллера.

0	0	0	0	0	1	1	1	1
1	0	0	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X

--> Диагностика внутреннего ОЗУ контроллера (класс 7, опкод 0). (*)

Команда проверяет правильность работы внутреннего ОЗУ контроллера. В случае ошибки генерируется ошибка с типом 3 и кодом 0.

0	1	1	1	0	0	0	0	0
1	0	0	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X

--> Тест устройства (класс 7, опкод 3).

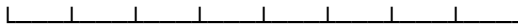
Команда проверяет состояние и работоспособность накопителя.

0	1	1	1	0	0	0	1	1
1	0	0	У	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	П	X	X	У	X	Ш	Ш	Ш

--> Расширенный тест контроллера (класс 0, опкод 4).

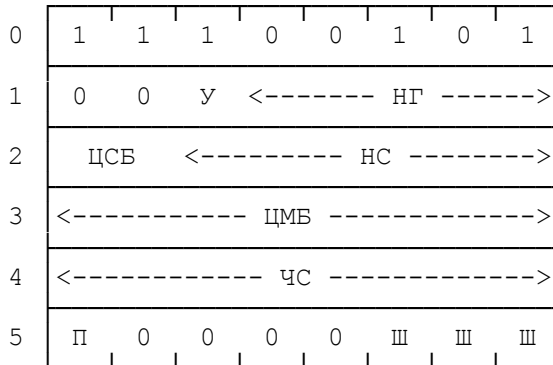
Команда проверяет функционирование составных частей контроллера: микропроцессора, ОЗУ, ПЗУ и схем коррекции ошибок.

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



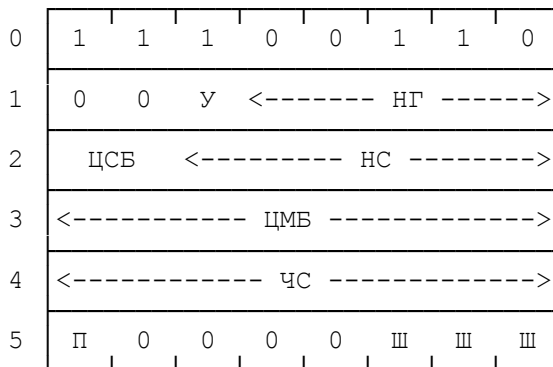
--> Считывание без контроля (класс 7, опкод 5).

Команда выполняет считывание поля данных одного сектора вместе с его контрольной суммой. Параметр "число секторов" игнорируется. При обнаружении ошибки контрольной суммы флаг ошибки не выставляется и коррекция не производится.



--> Запись без контроля (класс 7, опкод 6).

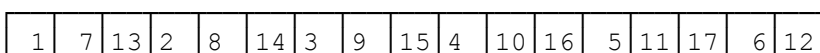
Команда выполняет запись поля данных одного сектора и его контрольной суммы, при этом логика коррекции ошибок не используется. Параметр "число секторов" не используется.



9.1.4. Разметка диска

Данные на жесткий диск записываются в секторах. Сектора располагаются на дорожках. Нумерация дорожек начинается с внешней стороны пластины (там расположена нулевая дорожка). Количество пластин (дисков) и головок, так же как и максимальное число дорожек, могут колебаться в довольно широких пределах и зависят от типа конкретного накопителя.

Дорожка обычно содержит от 8 до 26 секторов и для данного конкретного накопителя число секторов на дорожке постоянно. Начало дорожки определяется сигналом "индекс", который генерируется накопителем при каждом обороте диска. Далее следует первый сектор дорожки. Второй сектор будет отстоять от первого на число секторов, равное значению фактора чередования минус 1, третий еще на столько же и т. д. Таким образом при факторе чередования равном 3, сектора на 17-секторной дорожке будут располагаться следующим образом:



<---индекс	фактор чер-ния <----->	индекс -->
------------	------------------------------	------------

Формат сектора приведен в таблице.

Обозначение	Назначение	Длина (байт)
АМ ИНТ1	Адресный маркер Интервал	4 9-12
СИНХ1 ИНТ2 СРВ ЦСВ ЦМВ ГЛВ СЕКТ ФЛАГ НУЛЬ КС1	Поле идентификации: Синхробайт Интервал Байт для сравнения Старший байт номера цилиндра Младший байт номера цилиндра Номер головки Номер сектора Флаговый байт Нулевой байт Контрольная сумма поля идентификации	1 2 1 1 1 1 1 1 1 4
ИНТ3	Интервал	16
СИНХ2 ИНТ4 ДАнные КС2	Область данных: Синхробайт Интервал Поле данных Контрольная сумма поля данных	1 2 512 4
ИНТ5	Интервал	43

Начало сектора обозначается при помощи уникальной комбинации, называемой адресным маркером. Пять интервалов, обозначенных в таблице ИНТ1 - ИНТ5, заполняются нулями, причем длина ИНТ2 и ИНТ4 постоянна и равна 2 байтам, в то время как ИНТ1, ИНТ3 и ИНТ5 могут иметь разную длину, немного отличающуюся от значений, приведенных в таблице. Эти три интервала предназначены для подготовки накопителя к считыванию (записи) следующей за ним области, а различие в длине объясняется тем, что адресный маркер, поле идентификации и данные могут записываться в разное время поверх ранее имевшейся информации, что не обеспечивает точного совпадения физической длины записанной области и ранее имевшейся на этом месте области. Отличия в длине могут составлять несколько бит, чем объясняется наличие после после интервалов синхробайта, позволяющего определить истинную начальную границу области и правильно сгруппировать все последующие биты по байтам. Байт СРВ имеет постоянное значение для всех секторов и используется для проверки правильности считывания поля идентификации. Флаговый байт содержит служебную информацию, в частности отмечает дефектность сектора, если он не пригоден для записи. Контрольные суммы служат для проверки правильности считывания информации из поля идентификации и области данных и вычисляются специальным генератором кодов коррекции ошибок (Error Correction Code, ECC) с производящим полиномом 32-й степени.

9.2. Контроллер жестких дисков для АТ-подобных ПЭВМ

IBM PC AT отличается от всех предыдущих моделей IBM PC и совместимых ПЭВМ в следующем:

- стандартный BIOS обеспечивает возможность работы как с накопителями на гибких, так и с накопителями на жестких дисках;

- контроллеры жестких и гибких дисков расположены на одной плате;

- адреса портов ввода-вывода, предназначенных для управления жестким диском, и назначение портов полностью отличаются от XT-подобных ПЭВМ.

Возможно подключение к одному ПЭВМ двух жестких дисков. Каждый диск имеет свой набор портов (1F0h-1F7h для первого и 170h-177h для второго). Ниже будут описаны порты только первого диска. Назначение портов второго диска аналогично первому.

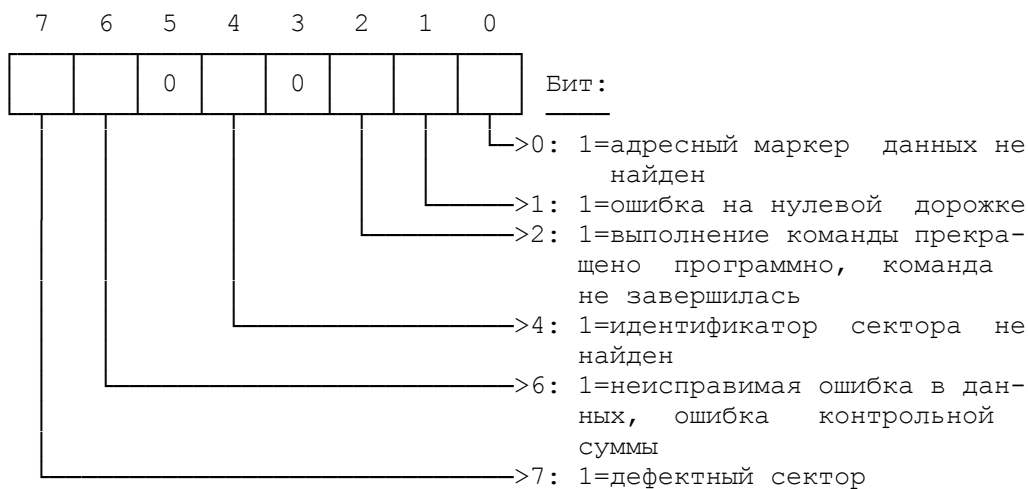
9.2.1. Описание портов ввода-вывода

Порт 1F0h

Предназначен для обмена данными с внутренним ОЗУ контроллера, являющимся промежуточным звеном между оперативной памятью ПЭВМ и накопителем.

Порт 1F1h.

При чтении через этот порт можно получить информацию о последней возникшей ошибке:



Порт 1F2h.

Чтение/запись числа секторов для последующей операции.

Порт 1F3h.

Чтение/запись параметра "номер сектора" для последующей операции.

Порт 1F4h.

Чтение/запись старших битов номера цилиндра для последующей операции (биты 0 - 1 - это биты 8 - 9 в 10-битовом номере цилиндра).

Порт 1F5h.

Чтение/запись младших восьми битов номера цилиндра для последующей операции.

Порт 1F6h.

Чтение/запись номера устройства и головки для последующей операции.

Порт 1F7h (запись).

Вывод команды по управлению накопителем.

Порт 1F7h (чтение).

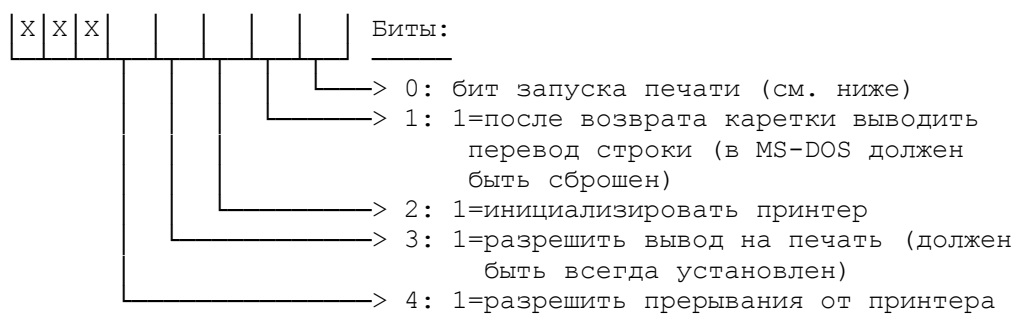
Чтение состояния накопителя и результатов предыдущей команды:



9.2.2. Система команд

Код команды заносится в порт 1F7h после того, как подготовлены все параметры в портах 1F2h - 1F6h. Выполнение команды начинается сразу после занесения в порт 1F7h кода команды. Команда состоит из одного байта. Старшие 4 бита содержат код требуемой операции, младшие содержат параметры. Перечень команд с возможными значениями параметров приведен в таблице.

Код операции	Назначение	Параметры
1	Рекалибровать (сброс головок на цилиндр 0)	Период импульсов сигнала "шаг"
2	Чтение секторов	Биты, определяющие необходимость повторных обращений к диску и повторных попыток считывания при ошибках контрольной суммы
3	Запись секторов	
4	Проверка секторов	



Примечания.

1. Для запуска печати символа нужно на короткое время установить бит 0 регистра управления, а затем сбросить его.

2. Прерывание происходит по окончании вывода символа на печать: для первого принтера на седьмом уровне контроллера прерываний (IRQ7, вектор прерывания 0Fh), для второго принтера - на пятом уровне (IRQ5, вектор 0Dh). Следует отметить также, что IRQ5 используется XT-контроллером жестких дисков для генерации своих прерываний. Обычно этот бит не используется (сбрасывается), а проверка готовности принтера производится на основании опроса регистра состояния.

Типичная последовательность действий для вывода на печать одного символа такова:

- вывести передаваемый байт в регистр данных;
- в цикле проверять состояние принтера до установки бита 7 регистра состояния (здесь возможно использование таймаута);
- проверить биты 3-5 регистра состояния на наличие ошибки;
- установить и сразу же сбросить нулевой бит регистра управления, для этого подходит следующая последовательность команд:

```

mov dx,37Ah          ;адрес регистра управления
mov al,00001101b    ;установить биты 0,2 и 3
out dx,al           ;вывести команду
xor al,1            ;сбросить бит 0
out dx,al           ;повторно вывести команду

```

- далее можно прочитать и запрограммировать регистр состояния и перейти к печати следующего символа.

Несмотря на простоту управления, принтер представляет из себя достаточно сложное устройство, имеющее свой микропроцессор, ПЗУ и ОЗУ. Микропроцессор используется для обработки управляющих последовательностей и управления печатью. ПЗУ содержит описание рисунков символов в зависимости от их кода и выбранной таблицы и программы на языке используемого микропроцессора. ОЗУ используется для временного хранения введенных данных и спроектированных пользователем символов. Ряд принтеров, помимо интерфейса Centronix, могут подключаться и к последовательному интерфейсу. Принтер обрабатывает ряд спецсимволов (таких, как звонок, забой, горизонтальная и вертикальная табуляция), а также имеет достаточно сложную систему команд для изменения режимов работы и управления печатью.

Обычно команды начинаются с символа ESC (шестнадцатиричный код 1Bh) и имеют длину от двух байт и более (существуют

командные последовательности, включающие по несколько килобайт информации, например команды загрузки спроектированных пользователем шрифтов). Спецсимволы и байты команд передаются принтеру так же, как и обычные символы, предназначенные для печати, описанным выше способом. Подобный способ управления удобен для программного изменения режимов работы устройства. Учитывая, что выпуском принтеров для ПЭВМ занимается достаточно большое число компаний и возможности принтеров постоянно совершенствуются, нет смысла приводить полностью какую-либо систему команд, так как, во-первых, она в разных марках и моделях принтеров различна, и, во-вторых, достаточно полно описывается в документации на каждый конкретный принтер.

Существует 2 основных стандарта на систему команд и таблицу используемых символов (стандарты IBM и EPSON), которых в основном придерживаются фирмы-изготовители, оставляя за собой право вносить в них изменения и дополнения. Единого стандарта как на систему команд, так и на расположение и рисунки символов второй половины таблицы ASCII (коды 129-255) нет. Наиболее часто используемые команды:

- установка вида шрифта и режима печати;
- черновой текстовый (draft) нормальной ширины (pica), узкий (condensed) или средней ширины (elita).
- качественный шрифт (NLQ, Proportional, Orator, Script и т. д.);
- выбор таблицы символов из ПЗУ (обычно имеется несколько таблиц, отличающихся расположением управляющих кодов, наличием и расположением псевдографических символов или курсива);
- загрузка в ОЗУ принтера спроектированных пользователем шрифтов и работа с ОЗУ принтера (именно в этой группе команд, имеются наибольшие различия между разными моделями);
- управление принтером в режиме точечной графики;
- установка служебных параметров (размеры отступов, страницы, шаг табуляции, расстояние между строками) и изменение характеристик (сброс, включение/отключение датчика конца бумаги, печать в одну или в две стороны и т. п.).

Ряд характеристик и режимов работы может выбираться как программно, так и аппаратно с использованием кнопок и клавиш на лицевой панели и DIP-переключателей. Существуют характеристики, изменить которые можно только аппаратно (например, возможность загрузки шрифтов или размер используемого ОЗУ).

11. Последовательный интерфейс RS-232

11.1. Общие сведения

В состав IBM PC могут входить до четырех последовательных интерфейсов, работающих в стандарте RS-232 (отечественный аналог - стык С2) и именуемых COM1 - COM4. Им выделены следующие адреса в области портов ввода-вывода:

COM1:	3F8h-3FFh	COM3:	338h-33Fh
COM2:	278h-2FFh	COM4:	238h-23Fh

(интерфейсы COM3 и COM4 поддерживаются только в моделях PS/2).

Каждый интерфейс связан с определенным уровнем контроллера прерываний:

- COM1 вызывает прерывание IRQ4 (Int 0Ch)
- COM2 вызывает прерывание IRQ3 (Int 0Bh)
- COM3 и COM4 не имеют стандартных векторов прерываний.

Каждое из устройств RS-232 представляет собой контроллер 8250, оснащенный 25- или 9-штырьковым разъемом на задней стенке корпуса ПЭВМ. Этот разъем может использоваться для подключения мыши, графопостроителя или организации связи между ПЭВМ. Контакты стыка RS-232 имеют следующие наименования

Название сигнала	Имя цепи		Номер контакта		Назначение	Направление
	EIA	CCITT	9-шт.	25-шт.		
DCD	CF	109	1	8	Связь модемов установлена	В ПЭВМ
RX	BB	104	2	3	Принимаемые данные	В ПЭВМ
TX	BA	103	3	2	Передаваемые данные	Из ПЭВМ
DTR	CD	108/2	4	20	Готовность ПЭВМ к работе	Из ПЭВМ
SG	AB	102	5	7	Сигнальная земля	—
DSR	CC	107	6	6	Готовность модема к работе	В ПЭВМ
RTS	CA	105	7	4	Запрос на передачу	Из ПЭВМ
CTS	CB	106	8	5	Готовность модема к передаче	В ПЭВМ
RI	CE	125	9	22	Индикатор вызова	В ПЭВМ
FG	AA	101	—	1	Защитная земля	—

Контроллер стыка RS-232 является полностью программируемым устройством; вы можете задать следующие параметры обмена: количество битов данных и стоп-битов, вид четности и скорость обмена в бодах (бит/с).

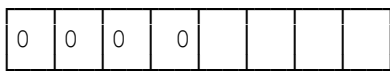
11.2. Описание портов

Ниже описаны порты ввода-вывода для COM1, имеющего базовый адрес 3F8h. Обратите внимание, что порты 3F8h и 3F9h имеют разное назначение в зависимости от бита 7 порта 3F8h (т. н. бит DLAB - Divisor Latch Access Bit).

Порт	Операция	Описание
3F8h	Запись	Регистр передатчика - сюда засылается байт для передачи
	Чтение	Регистр приемника - отсюда извлекается принятый байт
	Запись	Если DLAB=1, то сюда засылается младший байт скорости обмена (см. порт 3F9h)
3F9h	Запись	Если DLAB=1, то сюда засылается старший байт скорости обмена. Скорость задается значением делителя, равным $115200/V$, где V - скорость в бодах. Например, для скорости 9600 бод делитель равен $115200/9600=12=0Ch$, поэтому нужно вывести 0Ch в порт 3F8h и 0 в порт 3F9h.
	Запись	Регистр управления прерываниями (1 = разрешить)

прерывание) :

7 6 5 4 3 2 1 0

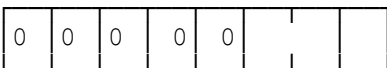


Бит:

- > 0: прерывание по приему символа
- > 1: прерывание по завершению передачи символа
- > 2: прерывание по обрыву линии или или ошибке в линии
- > 3: прерывание по изменению состояния модема (любой из линий CTS, DSR, RI и DCD)

3FAh Чтение Регистр идентификации прерывания. Когда произошло прерывание, здесь содержится причина, вызвавшая его:

7 6 5 4 3 2 1 0



Бит:

- > 0: 0=есть отложенные прерывания
- > 1-2: причина прерывания:
 - 11=ошибка или обрыв линии; сбрасывается чтением регистра состояния линии (порт 3FDh)
 - 10=принят символ; сбрасывается чтением приемника (порт 3F8h)
 - 01=передан символ; сбрасывается записью символа в регистр передатчика (порт 3F8h)
 - 00=изменение состояния модема (линий CTS, DSR, RI или DCD); сбрасывается чтением регистра состояния модема (порт 3FEh)

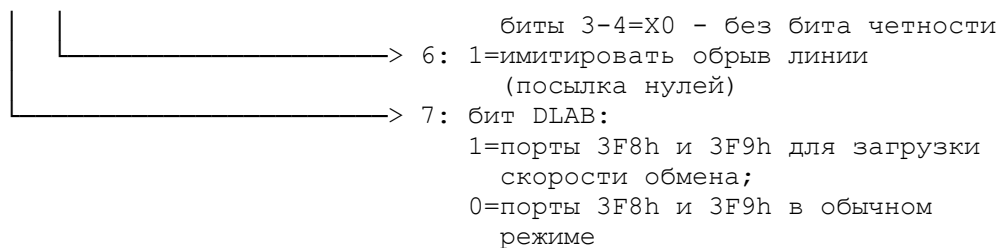
3FBh Чтение/ Запись Регистр управления линией:

7 6 5 4 3 2 1 0

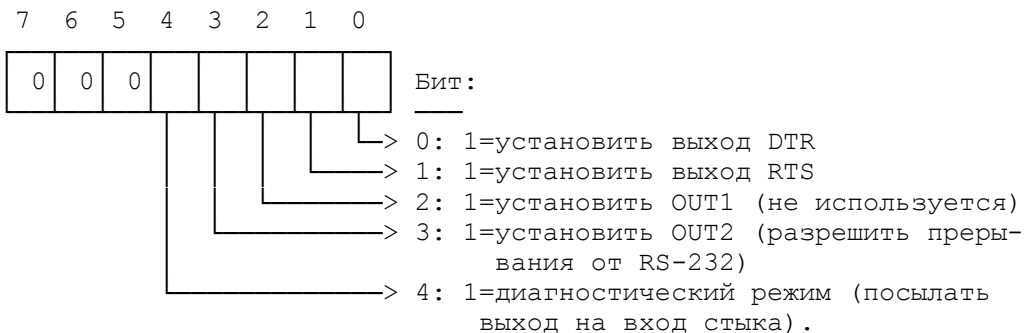


Биты:

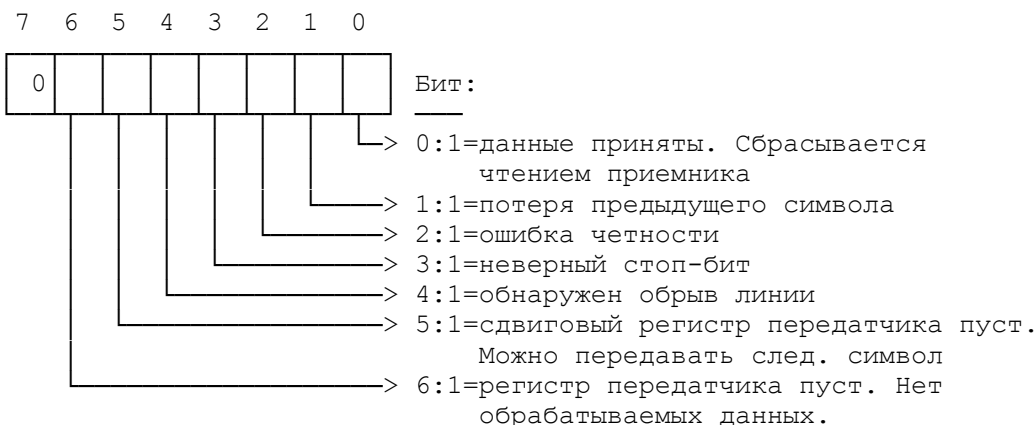
- > 0-1: число битов данных:
00=5, 01=6, 10=6, 11=8
- > 2: число стоп-битов:
0=1, 1=1.5 при 5 битах данных и 2 в противном случае
- > 3-4: тип четности:
X0=нет,
01=нечетная,
11=четная
- > 5: постоянная четность
0=отмена постоянной четности
1=постоянный бит четности, зависит от битов 3-4:
биты 3-4=01 - бит четности всегда 1
биты 3-4=11 - бит четности всегда 0



3FCh Запись Регистр управления модемом



3FDh Чтение Регистр состояния линии. Биты 1-4 вызывают прерывания по ошибке, если оно разрешено:



3FEh Чтение Регистр состояния модема. Биты 0-3 вызывают прерывание по изменению состояния модема, если оно разрешено:



Примечание: "Изменилось состояние линии..." означает, что данная линия стыка RS-232 изменила свое состояние по сравнению с последним чтением этого регистра.

11.3. Порядок инициализации 8250

Для подготовки контроллера 8250 к работе необходимо выполнить следующие шаги.

1. Установить бит DLAB порта 3FBh и заслать делитель, задающий скорость обмена, в порты 3F8h и 3F9h.
2. Инициализировать регистр управления линией (порт 3FBh); при этом сбросить бит DLAB.
3. Инициализировать регистр управления модемом (порт 3FCh).
4. Инициализировать регистр управления прерываниями (порт 3F9h) и, если прерывания разрешены, установить адрес программы обработки прерываний от стыка RS-232.

В качестве примера приведем набор подпрограмм, обеспечивающих обмен через порт COM1 в дуплексном режиме со скоростью 1200 бод.

```
title RS232
_DATA segment para public
Buf_Size equ 1024 ; размер буфера

Source db Buf_Size+2 dup (0) ; буфер приема символов
Src_ptr dw Source ; указатель позиции в буфере
Count dw 0 ; количество символов в буфере
Ser_ip dw 0 ; старый адрес Int 0Ch
Ser_cs dw 0
Save_ds dw 0 ; служебные переменные
Int_sts db 0
Overrun db 0
_DATA ends

_TEXT segment para public
assume cs:_TEXT, ds:_DATA
public Ser_Ini:near, Ser_Rst:near
public Get_Chr:near, Out_Chr:near
;
; Подпрограмма инициализации стыка COM1.
;
Ser_Ini proc near
push ax ; сохранить регистры
push dx
push bx
push es
in al,21h ; IMR 1-го контроллера прерываний
or al,10h ; запретить прерывание IRQ4 от COM1
out 21h,al
mov al,0Ch
mov ah,35h
int 21h ; взять вектор Int 0Ch в es:bx
mov Ser_ip,bx ; и сохранить его
mov Ser_cs,es
mov al,0Ch
mov dx,offset Ser_int
push ds
mov bx,cs
mov ds,bx
mov ah,25h
int 21h ; установить Int 0Ch = ds:dx
pop ds
pop es
```

```

pop    bx
cli    ; запретить прерывания
in     al,21h ; IMR 1-го контроллера прерываний
and    al,not 10h
out    21h,al ; разрешить прерывания от COM1
mov    dx,3FBh ; регистр управления линией
in     al,dx
or     al,80h ; установить бит DLAV
out    dx,al
mov    dx,3F8h
mov    al,60h
out    dx,al ; младший байт для скорости 1200 бод
inc    dx
mov    al,0
out    dx,al ; старший байт скорости
mov    dx,3FBh ; регистр управления линией
mov    al,00000011b ; 8 бит, 2 стоп-бита, без четности
out    dx,al
mov    dx,3F9h ; регистр разрешения прерываний
mov    al,1 ; разрешить прерывания по приему
out    dx,al
nop    ; и чуть-чуть подождать
nop
mov    dx,3FCh ; регистр управления модемом
mov    al,00001011b ; установить DTR, RTS и OUT2
out    dx,al
sti    ; разрешить прерывания
mov    dx,3F8h ; регистр данных
in     al,dx ; сбросить буфер приема
pop    dx
pop    ax
ret

```

Ser_Ini endp

```

;
; Подпрограмма отключения стыка COM1.
;

```

```

Ser_Rst proc near
push ax ; сохранить регистры
push dx

```

```

Wait_Free:
mov dx,3FDh ; регистр состояния линии
in al,dx
jmp short $+2 ; короткая задержка
test al,60h ; передача окончена?
jz Wait_Free ; ждем, если нет
mov dx,3F9h ; регистр разрешения прерываний
mov al,0 ; запретить прерывания
out dx,al
jmp short $+2 ; еще подождем...
jmp short $+2
mov dx,3FCh ; регистр управления модемом
mov al,00000011b ; активировать DTR и RTS
out dx,al
jmp short $+2
jmp short $+2
push bx
mov al,0Ch
mov dx,Ser_ip
push ds
mov bx,Ser_cs
mov ds,bx
mov ah,25h
int 21h ; восстановить вектор Int 0Ch
pop ds

```

```

    pop    bx
    cli                    ; запрет прерываний
    in     al,21h          ; читать маску прерываний
    jmp    short $+2
    or     al,10h          ; запретить IRQ4
    out    21h,al
    sti                    ; разрешение прерываний
    pop    dx
    pop    ax
    ret
Ser_Rst    endp
;
; 

|                                            |
|--------------------------------------------|
| Подпрограмма обработки прерываний от COM1. |
|--------------------------------------------|


;
Ser_Int    proc far
    push  ax
    push  dx
    push  ds
    mov   ax,seg _DATA
    mov   ds,ax
    mov   dx,3FAh        ; регистр идентификации прерываний
    in    al,dx
    mov   Int_Sts,al; сохраним его содержимое
    test  al,1           ; есть отложенные прерывания?
    jz    Is_Int         ; да
    pop   Save_ds       ; нет, передаем управление
    pop   dx             ; старому обработчику Int 0Ch
    pop   ax
    push  Ser_cs
    push  Ser_ip
    push  Save_ds
    pop   ds
    ret                 ; длинный переход
Is_Int:
    mov   al,64h        ; послать EOI для IRQ4
    out   20h,al        ; в 1-й контроллер прерываний
    test  Int_Sts,4     ; прерывание по приему?
    jnz   Read_Char    ; да
No_Char:
    sti                    ; нет, разрешить прерывания
    jmp   Int_Ret       ; и закончить обработку Int 0Ch
Read_Char:
    mov   dx,3FDh       ; регистр состояния линии
    in    al,dx
    and   al,2
    mov   Overrun,al; ovrrun<>0, если была потеря символа
    mov   dx,3F8h       ; регистр данных
    in    al,dx         ; вводим символ
    or    al,al         ; если принят нуль,
    jz    No_Char       ; то игнорируем его
    push  bx
    mov   ah,Overrun
    or    ah,ah         ; предыдущий символ потерян?
    jz    Save_Char    ; нет
    mov   ah,al         ; да,
    mov   al,7          ; заменяем его на звонок (07h)
Save_Char:
    mov   bx,Src_ptr; заносим символ в буфер
    mov   [bx],al
    inc   Src_ptr       ; и обновляем счетчики
    inc   bx
    cmp   bx,offset Src_ptr-2 ; если конец буфера
    jb   Ser_Int_1
    mov   Src_ptr,offset Source ; то "зацикливаем" на начало

```

```

Ser_Int_1:
    cmp     Count, Buf_Size ; буфер полон?
    jae     Ser_Int_2 ; да
    inc     Count          ; нет, учесть символ
Ser_Int_2:
    or      ah, ah         ; если была потеря символа
    jz      Ser_Int_3
    mov     al, ah         ; то занести в буфер сам символ
    xor     ah, ah
    jmp     short Save_Char
Ser_Int_3:
    pop     bx
    sti                                ; разрешить прерывания
Int_Ret:
    pop     ds
    pop     dx
    pop     ax
    iret
Ser_Int   endp

```

```

;
; Подпрограмма вывода символа AL в порт.
; При ошибке возвращает CF=1, иначе CF=0.
;

```

```

Out_Chr   proc near
    push  ax
    push  cx
    push  dx
    mov   ah, al
    sub   cx, cx
Wait_Line:
    mov   dx, 3FDh ; регистр состояния линии
    in    al, dx
    test  al, 20h  ; стык готов к передаче?
    jnz   Output  ; да
    jmp   short $+2
    jmp   short $+2
    loop  Wait_Line ; нет, ждем
    pop   dx
    pop   cx
    pop   ax
    stc                                ; нет готовности порта
    ret
Output:
    mov   al, ah
    mov   dx, 3F8h ; регистр данных
    jmp   short $+2
    out   dx, al   ; вывести символ
    pop   dx
    pop   cx
    pop   ax
    cld                                ; нормальный возврат
    ret
Out_Chr   endp

```

```

;
; Подпрограмма ввода символа из порта в AL.
; Если буфер пуст, возвращает CF=1, иначе CF=0.
;

```

```

Get_Chr   proc near
    cmp   Count, 0 ; буфер пуст?
    jne   loc_1729 ; нет
    stc                                ; да, возврат по ошибке
    ret
loc_1729:
    push  si

```



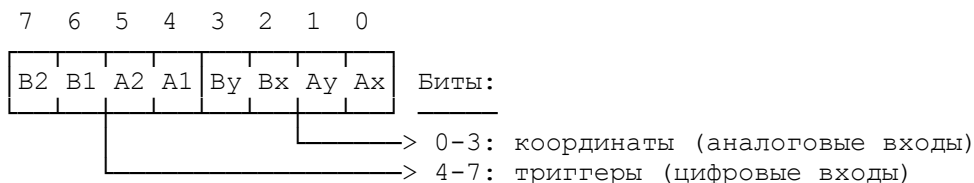
```

cli          ; запретим прерывания
mov  si,Src_ptr
sub  si,Count
cmp  si,offset Source
jae  loc_1730
add  si,Buf_Size
loc_1730:
mov  al,[si] ; выберем символ
dec  Count   ; и уменьшим счетчик
sti          ; разрешение прерываний
pop  si
clc          ; и нормальный возврат
ret
Get_Chr     endp
_TEXT      ends
end

```

12. Игровой адаптер

Игровой адаптер - это аналого-цифровой преобразователь, который может использоваться для подключения к ПЭВМ аналоговых устройств (например, джойстика или измерительных приборов). Он принимает до четырех цифровых входов типа "включено/выключено" (например, для нажатия кнопок) и до четырех аналоговых входов (таких, как координаты джойстика или показания электронного термометра). Игровой адаптер подключен в IBM PC к порту ввода 201h. Байт, введенный из этого порта, имеет следующий формат:



Для чтения триггеров используются команды:

```

mov  dx,201h
out  dx,al   ;инициировать обмен, AL -любой байт
in   al,dx   ;читать порт, биты 4-7 - 0=кнопка нажата
                        ;                               1=отпущена

```

Аналоговые входы считываются путем отслеживания времени, в течение которого соответствующий бит установлен в единицу, после команды OUT 201h,XXX. Пример чтения координаты X устройства A:

```

mov  dx,201h
out  dx,al   ;инициировать обмен, AL-любой байт
mov  cx,-1   ;счетчик для цикла
Again:
in   al,dx
inc  cx      ;увеличить счетчик
test al,1   ;бит AX сброшен?
jne  Again  ;нет, ждем

```

Результат в CX указывает координаты X. Этот способ зависит от тактовой частоты ЦП, поэтому аккуратней было бы использовать не счетчик цикла, а системный таймер.

13. CMOS

13.1. Доступ к CMOS

В состав IBM PC AT входят часы реального времени (RTC) и 64 байта неразрушающейся оперативной памяти (CMOS), питающиеся от автономного источника питания. При включении ПЭВМ содержимое CMOS анализируется BIOS, который извлекает из нее конфигурацию системы и текущие дату и время.

Для доступа к данным CMOS используются порты 70h и 71h. Чтение байта CMOS производится следующим образом:

```
mov    al,XXh
out    70h,al    ;выбрать адрес CMOS XX
jmp    $+2      ;небольшая задержка
in     al,71h    ;ввод байта из CMOS
```

Запись байта в CMOS производится аналогично:

```
mov    al,XXh
out    70h,al
jmp    $+2
in     al,71h
```

Адреса CMOS с 10h по 20h защищены контрольной суммой, хранящейся по адресу 2Eh. Поэтому изменения содержимого этих адресов необходимо сопровождать пересчетом и изменением контрольной суммы.

Порт 70h применяется не только для задания адреса CMOS, но и для разрешения или запрещения NMI (немаскируемого прерывания). Если бит 7 равен 0, то NMI разрешается, если 1 - запрещается, например:

```
mov    al,2Dh    ;2Dh -резервный адрес CMOS
out    70h,al    ;разрешить NMI
mov    al,0ADh
out    70h,al    ;запретить NMI
```

13.2. Содержимое CMOS

Краткая схема содержимого CMOS такова:

Адрес	Содержимое
00h-0Dh	Данные часов реального времени (RTC)
0Eh	Байт результатов начального тестирования
0Fh	Состояние перезагрузки
10h	Тип НГМД
11h	Резерв
12h	Тип жесткого диска
13h	Резерв
14h	Байт оборудования
15h-16h	Размер базовой памяти
17h-18h	Размер дополнительной памяти
19h-1Ah	Типы жестких дисков
1Bh-2Dh	Резерв
2Eh-2Fh	Контрольная сумма CMOS
30h-31h	Размер дополнительной памяти
32h	Текущее столетие
33h	Разная информация
34h-3Fh	Резерв

Приведем полное описание всех полей CMOS.

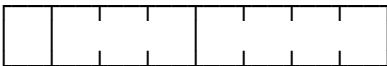
Адрес Описание

Примечание

0	Текущее время RTC, секунды	Все данные RTC
1	Время пробудки, секунды	хранятся в двоично-
2	Текущее время, минуты	десятичном формате
3	Время пробудки, минуты	(BCD)
4	Текущее время, часы	
5	Время пробудки, часы	
6	Текущий день недели	
7	Текущий день месяца	
8	Текущий месяц	
9	Текущий год	

0Ah Регистр A состояния RTC:

7 6 5 4 3 2 1 0



Биты:

- > 0-3: скорость отсчета (равна 0110)
- > 4-6: делитель (равен 010)
- > 7: флаг обновления (0=можно читать)

0Bh Регистр B состояния RTC:

7 6 5 4 3 2 1 0



Биты:

- > 0: тип времени (равен 0)
- > 1: 12- или 24-часовое время (равен 1=24 часа)
- > 2: формат данных: 0=BCD, 1=двоичный (равен 0)
- > 3: 1=разрешить прямоугольный импульс (равен 0)
- > 4: 1=разрешить прерывание по концу обновления (равен 0)
- > 5: 1=разрешить прерывание пробудки (равен 0)
- > 6: 1=разрешить периодические прерывания (равен 0)
- > 7: флаг обновления (0=можно читать)

0Ch Регистр C состояния RTC: биты состояния прерываний, только для чтения.

0Dh Регистр D состояния RTC. Бит 7=1, если CMOS получает питание; 0=нет питания от автономного источника.

0Eh Байт результатов начального тестирования:

7 6 5 4 3 2 1 0



Биты:

- > 2: 1=формат текущего времени и даты верен
- > 3: 1=жесткий диск не является загружаемым
- > 4: 1=размер ОЗУ неверен
- > 5: 1=оборудование не соответствует конфигурации CMOS
- > 6: 1=неверна контрольная сумма CMOS

_____> 7: 1=неисправен автономный источник питания CMOS

0Fh Байт состояния перезагрузки. Этот байт считывается после сброса ЦП, чтобы определить не был ли сброс вызван для вывода 80286 из защищенного режима. Он принимает значения:

- 0 = теплый рестарт (по Ctrl-Alt-Del) или неожиданный останов
- 1 = останов после определения размера ОЗУ
- 2 = останов после тестирования памяти
- 3 = останов по ошибке паритета памяти
- 4 = рестарт по запросу начального загрузчика
- 5 = рестарт по сбросу контроллера прерываний и JMP FAR PTR [0:467h]
- 6, 7, 8 = останов после теста защищенного режима
- 9 = рестарт по JMP FAR PTR [0:467h]

10h Тип НГМД А и В:

7 6 5 4 3 2 1 0

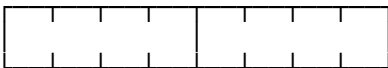


Биты:

- > 0-3: тип дисководов А
- > 4-7: тип дисководов В:
 - 0 = нет дисковода или его тип неизвестен
 - 1 = 360 К, 5 1/4 дюйма
 - 2 = 1.2 М, 5 1/4 дюйма
 - 3 = 720 К, 3 1/2 дюйма

12h Тип жестких дисков С и D:

7 6 5 4 3 2 1 0

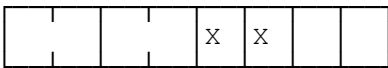


Биты:

- > 0-3: тип дисководов С
- > 4-7: тип дисководов D:
 - 0000 = диска нет
 - 1111 = см. адреса 19h и 1Ah
 - прочие=тип диска (см. приложение 2)

14h Байт оборудования:

7 6 5 4 3 2 1 0



Биты:

- > 0: 1=есть НГМД
- > 1: 1=есть сопроцессор 80287
- > 4-5: тип дисплея:
 - 00 - EGA или VGA
 - 01 - CGA, 40 столбцов
 - 10 - CGA, 80 столбцов
 - 11 - монохромный
- > 6-7: количество НГМД - 1

15h-16h Размер базовой памяти (0100h=256 К, 0200h=512 К, 0280h=640 К)

17h-18h Размер дополнительной памяти, Кб (от 0 до 3С00h)

19h	Тип диска C, если его тип в 12h равен 0Fh
1Ah	Тип диска D, если его тип в 12h равен 0Fh
2Eh-2Fh	Контрольная сумма адресов CMOS с 10h по 20h (обычная 16-битовая сумма этих байтов; 2Eh содержит старший байт КС)
30h-31h	Размер дополнительной памяти, КБ (0-3C00h), определенный при начальном тестировании
32h	Двоично-десятичный номер столетия
33h	Разная информация. Бит 7: 1=установлена опция 128К ОЗУ Бит 6 используется программой SETUP

14. Прочие порты

В этом разделе приведено описание портов, не относящихся к конкретным устройствам.

Адрес	Операция	Назначение
80h	запись	Используется в АТ на этапе начального тестирования для накопления информации об ошибках.
90h	запись	Управляющий порт центрального арбитра PS/2
91h	чтение	Обратная связь с выбранной платой PS/2
92h	запись	Управляющий порт А PS/2
94h	запись	Установка параметров системной платы PS/2
96h, 97h	запись	Выбор коннектора канаклов в PS/2
0A0h	запись	Разрешение или запрещение NMI в PC/XT: out 0A0h, 80h - разрешает NMI; out 0A0h, 0 - запрещает NMI.
0F0h	запись	Вывод любого байта сбрасывает защелку сопроцессора 80287, которая включается по незамаскированной ошибке сопроцессора.
0F1h	запись	Вывод любого байта сбрасывает сопроцессор 80287 из защищенного режима в реальный.
0F8h-0FCh	—	Эти порты используются процессором 80286 для обмена данными с сопроцессором.
210h	запись	Управляющий порт блока расширения XT.
213h	запись	Разрешение блока расширения XT.
215h	запись	Регистр шины данных блока расширения XT.
215h	запись	Регистр адреса блока расширения XT (старший байт).
216h	запись	Регистр адреса блока расширения XT (младший байт).

Всякий раз, когда клавиатура вызывает прерывание по нажатию или отпусканю клавиши, она передает обработчику этого прерывания "скан-код" данной клавиши, т. е. ее порядковый номер на клавиатуре. При этом скан-код клавиши является семибитовым, а старший бит байта, содержащего скан-код, сообщает причину прерывания: 0 - клавиша нажата, 1 - клавиша отпущена.

Скан-код		Клавиша	Скан-код		Клавиша
01h	1	Esc	2Bh	43	\
02h	2	1 !	2Ch	44	Z
03h	3	2 @	2Dh	45	X
04h	4	3 #	2Eh	46	C
05h	5	4 \$	2Fh	47	V
06h	6	5 %	30h	48	B
07h	7	6 ^	31h	49	N
08h	8	7 &	32h	50	M
09h	9	8 *	33h	51	, <
0Ah	10	9 (34h	52	. >
0Bh	11	0)	35h	53	/ ?
0Ch	12	- _	36h	54	правый Shift
0Dh	13	= +	37h	55	PrtSc *
0Eh	14	BackSpace	38h	56	Alt
0Fh	15	Tab	39h	57	пробел
10h	16	Q	3Ah	58	CapsLock
11h	17	W	3Bh	59	F1
12h	18	E	3Ch	60	F2
13h	19	R	3Dh	61	F3
14h	20	T	3Eh	62	F4
15h	21	Y	3Fh	63	F5
16h	22	U	40h	64	F6
17h	23	I	41h	65	F7
18h	24	O	42h	66	F8
19h	25	P	43h	67	F9
1Ah	26	[{	44h	68	F10
1Bh	27] }	45h	69	NumLock
1Ch	28	Enter	46h	70	ScrollLock
1Dh	29	Ctrl	47h	71	Home [7]
1Eh	30	A	48h	72	стр. вверх [8]
1Fh	31	S	49h	73	PgUp [9]
20h	32	D	4Ah	74	-
21h	33	F	4Bh	75	<- [4]
22h	34	G	4Ch	76	[5]
23h	35	H	4Dh	77	-> [6]
24h	36	J	4Eh	78	+
25h	37	K	4Fh	79	End [1]
26h	38	L	50h	80	стр. вниз [2]
27h	39	; :	51h	81	PgDn [3]
28h	40	' "	52h	82	Ins [0]
29h	41	` ~	53h	83	Del [.]
2Ah	42	левый Shift	54h	84	SysReq

Приложение 2. Типы жестких дисков

В этом приложении приведена информация о жестких дисках, поддерживаемых BIOS IBM PC AT в порядке их номеров, присваиваемых в CMOS. Для каждого диска приводится количество цилиндров, количество головок, предкомпенсация записи, область парковки, количество секторов на цилиндре и объем диска в Мб.

Номер типа	Кол-во цилиндров	Кол-во головок	Предкомп. записи	Область парковки	Кол-во секторов	Объем диска, Мб
1	306	4	128	305	17	10,1
2	615	4	300	615	17	20,4
3	615	6	300	615	17	30,6
4	940	8	512	940	17	62,4
5	940	6	512	940	17	46,8
6	615	4	нет	615	17	20,4
7	462	8	256	511	17	30,6
8	733	5	нет	733	17	30,4
9	900	15	нет	901	17	112,0
10	820	3	нет	820	17	20,4
11	855	5	нет	855	17	35,4
12	855	7	нет	855	17	49,6
13	306	8	128	319	17	20,3
14	733	7	нет	733	17	42,5
15		Зарезервирован - см. 13.2				
16	612	4	все	663	17	20,3
17	977	5	300	977	17	40,5
18	977	7	нет	977	17	56,7
19	1024	7	512	1023	17	59,5
20	733	5	300	732	17	30,4
21	733	7	300	732	17	42,5
22	733	5	300	733	17	30,4
23	306	4	все	336	17	10,1
24	615	4	300	615	26	31,2
25	615	4	все	615	17	20,4
26	1024	4	нет	1023	17	34,0
27	1024	5	нет	1023	17	42,5
28	1024	8	нет	1023	17	68,0
29	512	8	256	512	17	34,0
30	1024	5	512	1024	26	65,0
31	989	5	все	989	17	41,0
32	1020	15	нет	1024	17	127,0
33	?	?	?	?	?	?
34	?	?	?	?	?	?
35	1024	9	1024	1024	17	76,5
36	1024	5	512	1024	17	42,5
37	830	10	нет	830	17	68,8
38	823	10	256	824	17	68,3
39	615	4	128	664	17	20,4
40	615	8	128	664	17	40,8
41	917	15	нет	918	17	114,1
42	1023	15	нет	1024	17	127,3
43	823	10	512	823	17	68,3
44	820	6	нет	820	17	40,8
45	1024	8	нет	1024	17	68,0
46	925	9	нет	925	17	69,1
47	699	7	256	700	17	40,6

Примечания.

1. Тип 1 - это жесткий диск первых РС XT, тип 2 - стандартный жесткий диск первых РС AT.

2. Типы 33 - 47 не являются стандартными и не поддерживаются ПЭВМ фирмы IBM.

Приложение 3. Подготовка ПЭВМ к работе

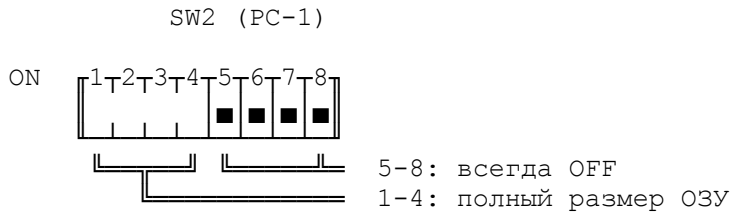
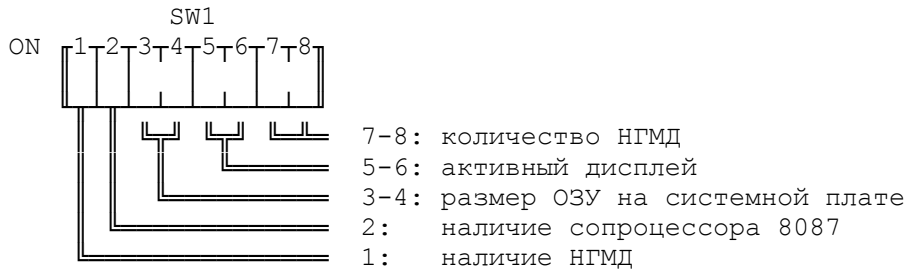
В этом приложении описана установка DIP-переключателей на системной плате ПЭВМ и адаптере EGA-дисплеев. Обычно ЭВМ

поступает заказчику уже с установленной конфигурацией, но при изменении оборудования (например, при установке большей памяти или дополнительных дисководов) может потребоваться изменение переключателей.

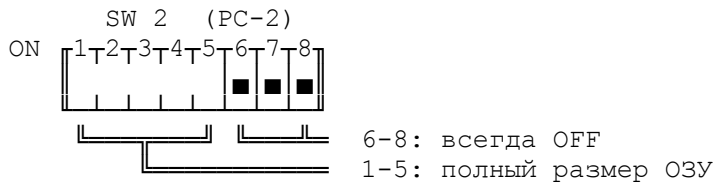
Приведенное здесь описание относится к ПЭВМ фирмы IBM; компьютеры других фирм могут иметь другие стандарты. Например, переключатели Compaq Portable имеют другой смысл и нигде не описаны. Переключатели Deskpro подписаны внутри корпуса ЭВМ.

Переключатели IBM PC

Первоначальная модель IBM PC, имевшая 64К ОЗУ на системной плате (далее называемая PC-1), настраивается двумя колодками переключателей, помеченными SW1 и SW2.

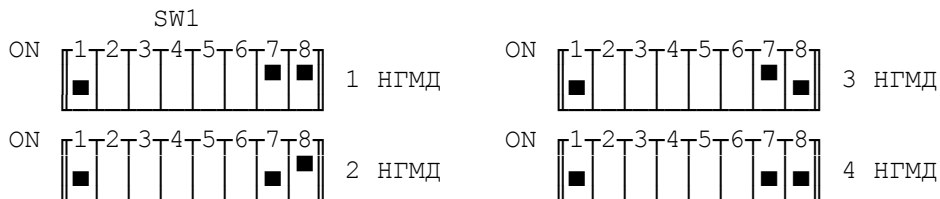


Более поздняя версия IBM PC (называемая здесь PC-2) интерпретирует переключатели несколько иначе. Точнее говоря, SW1 имеет то же назначение, а SW2 показано ниже:



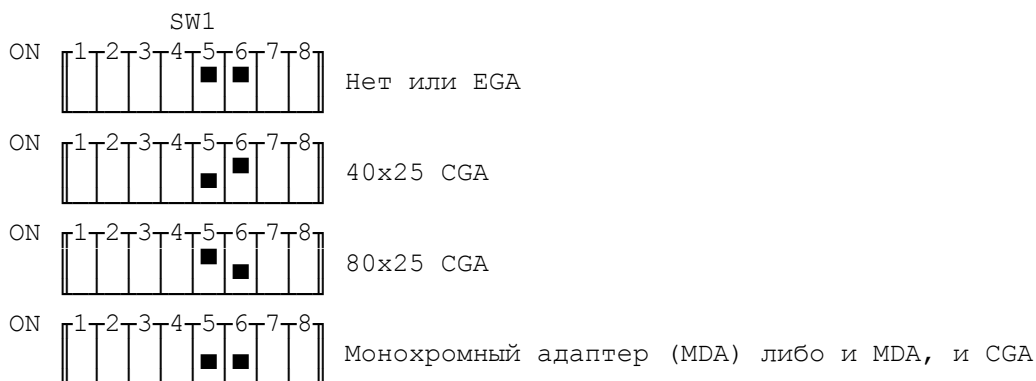
Опишем подробно назначение переключателей.

Количество накопителей на гибких дисках (НГМД):

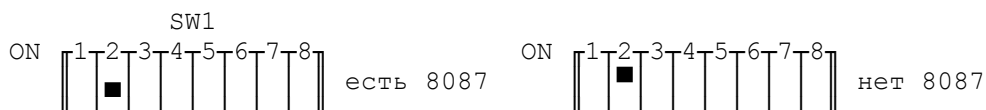


Примечание. Ключ 1 в позиции ON означает отсутствие НГМД.

Активный дисплей:



Наличие сопроцессора 8087:

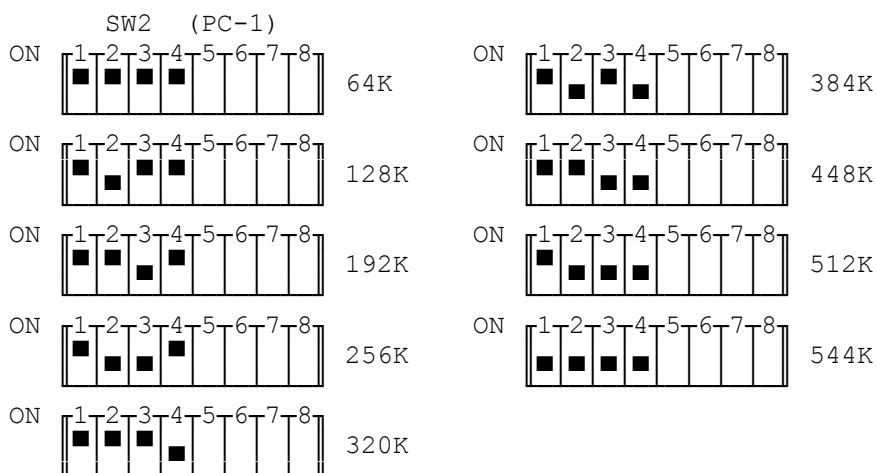


Размер ОЗУ на системной плате:



Полный размер ОЗУ (SW2 для PC-1):

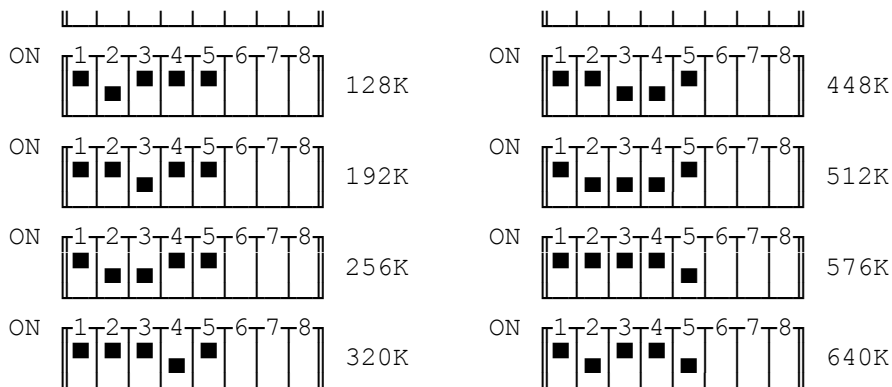
Для PC-1 ключи 5 - 8 всегда в положении OFF. Максимальный размер ОЗУ - 544К.



Полный размер ОЗУ (SW2 для PC-2):

Для PC-2 ключи 6 - 8 всегда в положении OFF. Максимальный размер ОЗУ - 640К. Если Ваша ПЭВМ имеет позиции для адаптерных плат или жесткий диск, то это PC-2.





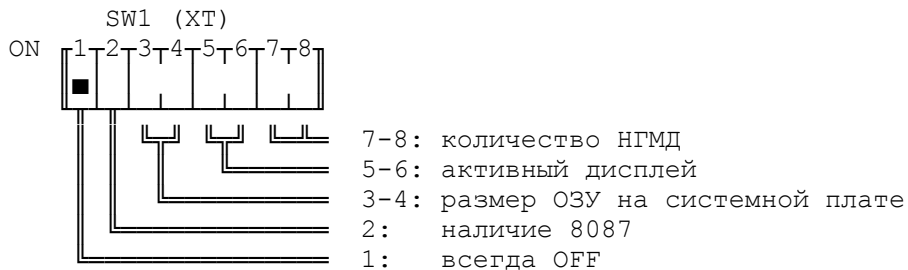
Примечания:

1. Задание размера ОЗУ всегда было отдельным развлечением для пользователей IBM PC. Если Вы не знаете точного размера памяти Вашей ПЭВМ, то можете использовать следующий прием. Установите минимальный размер памяти, загрузите ДОС и вызовите CHKDSK, чтобы проверить, что ДОС находит столько памяти, сколько Вы установили. Затем увеличивайте размер ОЗУ до тех пор, пока не произойдет ошибка.

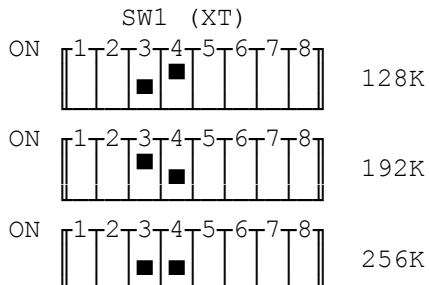
2. Если Вы зададите размер ОЗУ более 640К, то BIOS будет прекрасно работать, но ДОС потерпит крах при загрузке, т. к. часть COMMAND.COM грузится в старшие адреса памяти.

Переключатели IBM PC XT

IBM PC XT имеет одну колодку переключателей (SW1):



Ключи 2, 5-6 и 7-8 имеют тот же смысл, что на IBM PC. Ключи 3-4 задают размер ОЗУ на системной плате:

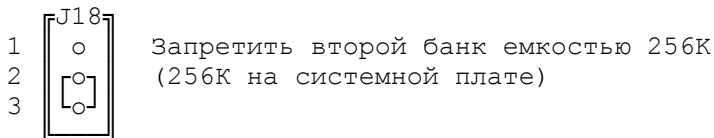
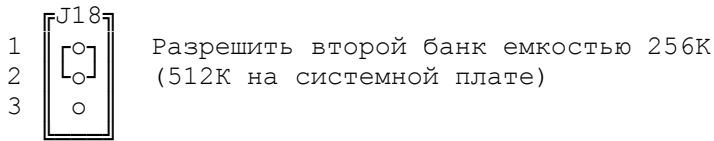


Примечание. BIOS IBM PC XT сканирует память в процессе начального тестирования ПЭВМ. Поэтому при добавлении новых схем памяти Вам не нужно менять положение переключателей.

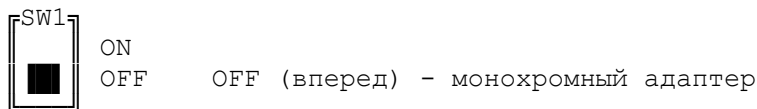
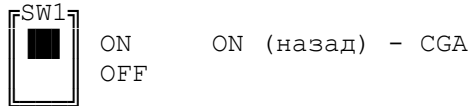
Переключатели IBM PC AT

IBM PC AT не имеет DIP-переключателей. Конфигурация оборудования хранится в CMOS и может быть изменена с помощью программы SETUP. Описание CMOS см. в п. 13.

В передней части системной платы имеется переключатель J18, управляющая использованием ОЗУ на системной плате:



Еще один переключатель, помеченный SW1, находится в задней части системной платы и задает первичный дисплей ПЭВМ:

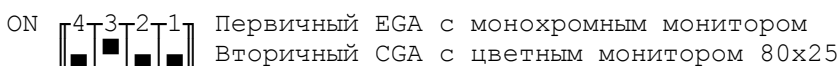
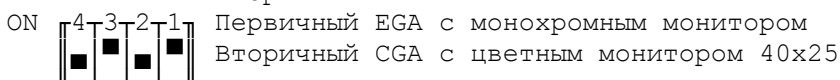
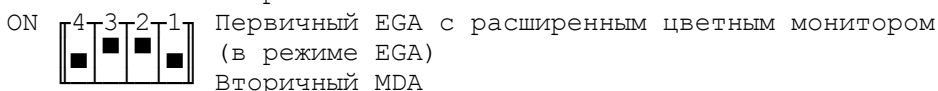
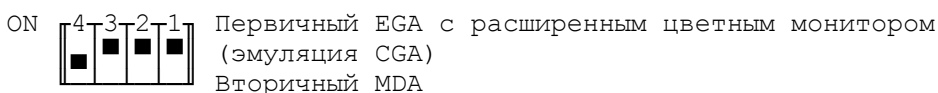
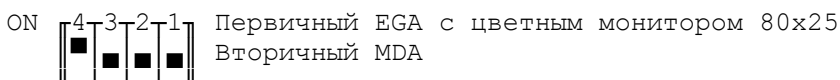
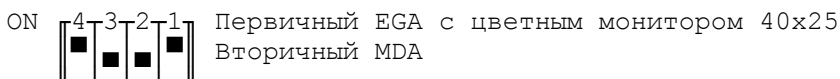


Примечание. Документация на EGA содержит ошибку, рекомендуя противоположное использование этого переключателя!

Переключатели адаптера EGA

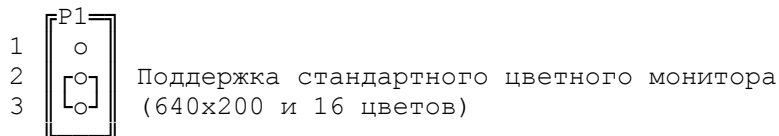
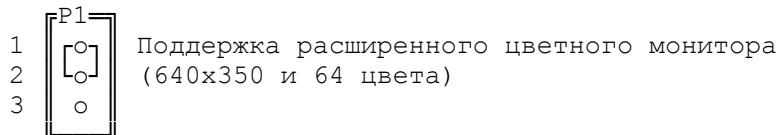
DIP-переключатели на адаптере EGA указывают тип дисплея, подключенного к адаптеру, и наличие других дисплейных адаптеров. Ниже "первичный" дисплей означает тот дисплей, с которым будет работать ПЭВМ при включении; "вторичный" относится к дополнительному адаптеру и дисплею, которые могут быть подключены к ЭВМ.

Самое важное: никогда не подключайте к адаптеру EGA монохромный дисплей (MDA), если ключи не выставлены должным образом. Вы можете физически разрушить дисплей!

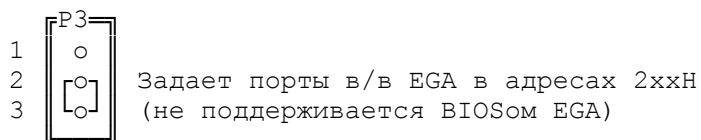
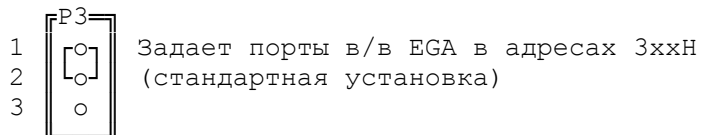


ON		Первичный MDA Вторичный EGA с цветным монитором 40x25
ON		Первичный MDA Вторичный EGA с цветным монитором 80x25
ON		Первичный MDA Вторичный EGA с расширенным цветным монитором (эмуляция CGA)
ON		Первичный MDA Вторичный EGA с расширенным цветным монитором (в режиме EGA)
ON		Первичный CGA с цветным монитором 40x25 Вторичный EGA с монохромным монитором
ON		Первичный CGA с цветным монитором 80x25 Вторичный EGA с монохромным монитором

EGA имеет три переключки (P1 - P3):



P2 (не показанный здесь) - это коннектор светового пера.



Глоссарий

Адаптер (adapter).
См. контроллер.

Ассемблер (assembly language).
Язык программирования на уровне команд центрального процессора.

Байт (byte).
Минимальная адресуемая единица памяти ЭВМ размером 8 бит.

Бит (bit).
Двоичный разряд, принимающий значения 0 или 1.

Блок (block).

Совокупность данных, которые вводятся или выводятся за одну операцию ввода-вывода.

Дисплей (display).

Устройство отображения символьной и графической информации.

Интерфейс (interface).

Аппаратное сопряжение двух устройств, обеспечивающее их электрическое и логическое согласование.

Интерфейс параллельный (parallel interface).

Сопряжение ПЭВМ с внешним устройством, при котором байт информации передается одновременно по восьми линиям. Обычно используется для подключения печатающих устройств.

Интерфейс последовательный (serial interface);

Сопряжение ПЭВМ с внешним устройством, при котором информация передается побитно. Используется для подключения мыши, графопостроителя и для межмашинной связи.

Канал (channel).

Устройство или независимая часть устройства, используемое для передачи данных между ЭВМ и периферией.

Контроллер (controller).

Микропроцессорное устройство, обеспечивающее связь ЭВМ с периферийным устройством.

Накопитель дисковый (disk drive).

Устройство внешней памяти на магнитных дисках, подключается к ЭВМ через контроллер. Диски ПЭВМ делятся на гибкие (дискеты) и жесткие (винчестерские).

Память оперативная, ОЗУ (random access memory, RAM).

Часть ЭВМ, используемая для хранения данных в процессе работы ЭВМ. При выключении ЭВМ содержимое ОЗУ теряется.

Память постоянная, ПЗУ (read-only memory, ROM).

Часть ЭВМ, используемая для хранения неизменяемой информации и доступная только для чтения.

Порт ввода-вывода (i/o port).

Адрес в пространстве ввода-вывода, обеспечивающий доступ к связанному с данным портом регистром внешнего устройства.

Позиции адаптерных плат (expansion slots).

Свободные позиции в корпусе ЭВМ, в которые могут быть вставлены платы контроллеров устройств ввода-вывода.

Прерывание (interrupt).

Внешнее событие, вызывающее прерывание работы ЦП и передачу управления программе обработки данного прерывания.

Прямой доступ в память, ПДП (direct memory access, DMA).

Способ доступа к памяти ЭВМ, при котором периферийный процессор работает с памятью, "воруя" циклы шины у центрального процессора.

Регистры (registers).

Аппаратные компоненты ЦП, предназначенные для обработки данных с высокой скоростью.

Слово (word).

Адресуемая единица машинной памяти, состоящая из двух

соседних байтов.

Центральный процессор, ЦП (central processing unit, CPU).

Основное устройство ЭВМ, осуществляющее выборку, декодирование и исполнение команд, а также управление периферийными устройствами через системную шину.

DIP-переключатели (DIP-switchs).

Переключатели на контактных колодках плат, позволяющие изменить конфигурацию соответствующего устройства.

□