

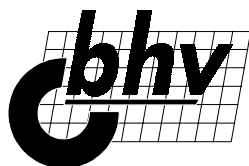
ПРОГРАММИРОВАНИЕ SVGA-ГРАФИКИ для IBM PC

- Особенности режимов SVGA
- Программирование рисунков и цветов
- Спецэффекты



Павел Соколенко

Программирование SVGА-ГРАФИКИ для IBM PC



Санкт-Петербург

Дюссельдорф ♦ Киев ♦ Москва ♦ Санкт-Петербург

В книге изложены основы программирования компьютерной графики для IBM PC на языке ассемблера. В ней рассмотрены: особенности основных видеорежимов SVGA, программирование построения рисунков и палитры цветов, работа с курсором и мышью, вывод текстовых сообщений и получение спецэффектов. Излагаемый материал иллюстрируется многочисленными примерами. В приложениях вы найдете описание графического стандарта BMP, работы с оперативной памятью компьютера, а также техники составления подпрограмм для алгоритмических языков высокого уровня.

Книга предназначена для читателей, интересующихся программированием компьютерной графики, и может быть рекомендована как начинающим, так и опытным программистам.

Для программистов

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Руководитель проекта	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Наталья Таркова</i>
Редактор	<i>Михаил Кокорев</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Игоря Цырульниковой</i>
Зав. производством	<i>Николай Тверских</i>

Соколенко П. Т.

Программирование SVGA-графики для IBM PC. — СПб.: БХВ-Петербург, 2001. — 432 с.

ISBN 5-94157-127-5

© П. Т. Соколенко, 2001

© Оформление, издательство "БХВ-Петербург", 2001

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 06.09.01.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 34,83.

Тираж 3000 экз. Заказ

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар, № 77.99.1.953.П.950.3.99 от 01.03.1999 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН.
199034, Санкт-Петербург, 9-я линия, 12.

Содержание

Предисловие	7
Глава 1. Видеосистемы и стандарт VESA	9
1.1. Видеосистемы и их стандартизация.....	10
1.1.1. Мониторы.....	10
1.1.2. Видеокарты и стандарты.....	11
1.1.3. Акселераторы.....	16
1.2. Общая характеристика стандарта VESA.....	21
1.2.1. Стандартизация видеорежимов.....	21
1.2.2. Информационные функции VBE.....	27
1.2.3. Основные функции VBE 1.2.....	32
1.2.4. Новые возможности VBE 2.0.....	38
Глава 2. Особенности работы в режимах VESA	43
2.1. Проверка поддержки видеорежима.....	44
2.2. Обработка информации о режиме.....	50
2.3. Процедуры для работы с одним окном видеопамяти.....	52
2.4. Работа с двумя окнами видеопамяти.....	57
2.5. Страничная организация видеопамяти.....	60
2.6. Часто используемые в примерах имена.....	64
2.7. Раздел для начинающих.....	69
Глава 3. Видеорежимы packed pixel graphics	77
3.1. Работа с отдельными точками.....	78
3.1.1. Команды для манипуляции с точками.....	78
3.1.2. Окна видеопамяти.....	82
3.1.3. Точки и их адреса.....	86
3.2. Построение геометрических фигур.....	90
3.2.1. Прямые линии.....	90
3.2.2. Прямоугольники.....	98
3.3. Построение рисунков.....	103
3.3.1. Варианты построения строк.....	103
3.3.2. Воспроизведение не сжатых рисунков.....	111
3.3.3. Воспроизведение сжатых рисунков.....	117
3.3.4. Заключительные замечания.....	125

Глава 4. Цвет на экране	129
4.1. Как получается цвет точки.....	129
4.2. Исходная цветовая палитра.....	132
4.3. Функции BIOS.....	137
4.4. Простая установка палитры	142
4.5. Манипуляции с палитрой цветов	148
Глава 5. Работа с текстом.....	157
5.1. Текстовые режимы	157
5.1.1. Русский текст на экране.....	158
5.1.2. Общая характеристика процесса вывода текста.....	161
5.1.3. Вывод текста с использованием поддержки DOS и BIOS.....	167
5.1.4. Непосредственная работа с видеобufferом.....	172
5.2. Графические режимы.....	178
5.2.1. Таблицы символов	178
5.2.2. Программный знакогенератор	181
5.2.3. Вывод информационных строк	187
5.2.4. Текстовый курсор в графическом режиме	193
5.2.5. Ввод символов с клавиатуры.....	201
Глава 6. Курсор и мышь	207
6.1. Построение рисунка курсора	207
6.1.1. Курсоры для Windows.....	208
6.1.2. Предварительная подготовка рисунка.....	210
6.1.3. Немамаскируемый курсор.....	214
6.1.4. Маскируемый курсор.....	218
6.1.5. Замечания к описанным подпрограммам.....	223
6.2. Подготовка к работе с манипулятором "мышь"	226
6.2.1. Общее описание драйвера мыши	227
6.2.2. Предварительные действия	231
6.3. Работа в режиме опроса драйвера мыши.....	237
6.3.1. Управляющий алгоритм для режима опроса.....	237
6.3.2. Формирование кодов событий	241
6.3.3. Управление перемещением курсора.....	246
6.4. Работа в режиме прерываний	249
6.4.1. Функции драйвера	250
6.4.2. Примеры прерывающих подпрограмм.....	254
Глава 7. Цвет в коде точки	261
7.1. Кодирование цвета.....	261
7.1.1. Среднее количество цветов.....	262
7.1.2. Максимальное цветовое разрешение	263
7.1.3. 24-разрядный код точки.....	265
7.2. Координаты и адреса точек.....	269
7.3. Линии, строки и прямоугольные области.....	275
7.3.1. Подпрограммы для рисования линий.....	275
7.3.2. Подпрограммы для построения строк.....	282
7.3.3. Работа с прямоугольными областями.....	286

7.4. Рисунки, использующие палитру	291
7.4.1. Преобразование палитры в форматы Hi-Color.....	292
7.4.2. Преобразование палитры в форматы True Color	296
7.4.3. Построение рисунков с использованием палитры	298
7.5. Рисунки, не использующие палитру	303
7.5.1. Рисунки, подготовленные в стандарте BMP	303
7.5.2. Рисунки, подготовленные в стандарте PCX.....	308
7.5.3. Способы сжатия полноцветных рисунков.....	313
7.6. Наложение рисунков и спецэффекты.....	319

Приложение А. Рисунки в файлах BMP..... 331

A.1. Общая характеристика стандарта.....	332
A.1.1. Заголовок файла для Windows.....	332
A.1.2. Заголовок файла для OS/2.....	333
A.1.3. Образ рисунка в файле.....	334
A.2. Общая схема обработки заголовка файла	336
A.2.1. Возможные отклонения от стандарта.....	337
A.2.2. Ввод спецификации и открытие файла	337
A.2.3. Чтение заголовка файла и палитры.....	340
A.2.4. Анализ основных полей заголовка	341
A.2.5. Манипуляции с палитрой.....	345
A.3. Построение рисунков, использующих палитру.....	350
A.3.1. Построение рисунка сверху вниз.....	350
A.3.2. Построение рисунка снизу вверх.....	353
A.3.3. Универсальная процедура построения рисунка	355

Приложение Б. Оперативная память 361

B.1. Обычная память (Conventional Memory).....	361
B.1.1. Сегменты оперативной памяти.....	362
B.1.2. Сегментирование текстов программ.....	364
B.1.3. Динамическое управление памятью	368
B.1.4. Использование функций DOS	372
B.2. Расширенная память (Expanded Memory).....	376
B.2.1. Спецификация расширенной памяти	376
B.2.2. Использование функций драйвера	380
B.2.3. Работа с расширенной памятью.....	383
B.3. Расширенная память (Extended Memory).....	388

Приложение В. Оформление подпрограмм..... 393

V.1. Классификация подпрограмм	393
V.2. Оформление программных модулей.....	397
V.3. Параметры в стеке	403
V.4. Работа процедур со стеком.....	409
V.5. Учет особенностей компилятора.....	414

Список литературы 423

Предметный указатель..... 425

Предисловие

Основы компьютерной графики были заложены еще на больших ЭВМ, задолго до появления персональных компьютеров. Ее первые практические применения были связаны с решением задач из области автоматизации проектирования архитектурных и инженерно-технических сооружений.

Массовое распространение и непрерывное совершенствование технических характеристик персональных компьютеров и периферийного оборудования способствовало расширению круга задач, при решении которых используется графика. В свою очередь, развитие и усложнение графики стимулирует создание все более совершенного компьютерного видеоборудования. Кроме того, непрерывно расширяется круг специалистов, вовлеченных в программирование и использование графических приложений. Поэтому литература, посвященная различным аспектам видеографики, пользуется постоянным спросом и нуждается в периодическом обновлении. Однако большинство публикуемых книг являются руководствами для пользователей, а не для программистов, поскольку содержат описание техники работы с различными графическими редакторами. Автор решил попытаться восполнить этот пробел и написать руководство, содержащее описание способов программирования базовых элементов компьютерной графики и получения спецэффектов.

В свое время на русский язык было переведено и опубликовано несколько обстоятельных руководств по программированию видеосистем для IBM PC. Технический прогресс беспощаден и они устарели вместе с описанными видеосистемами. Предлагаемая книга продолжает тему программирования видеосистем, но уже современного образца, с использованием стандарта VBE, разработанного ассоциацией VESA. Этот стандарт создавался специально для того, чтобы программирование графических объектов не зависело от особенностей видеокарт, выпускаемых различными фирмами.

В соответствии с темой книги в качестве языка программирования выбран ассемблер для IBM PC. Можно до хрипоты спорить о достоинствах и недос-

татках ассемблера, но всегда остается класс задач, которые имеет смысл программировать только на этом языке, и всегда остается категория программистов, которые обязаны в совершенстве владеть ассемблером. В первую очередь именно им адресована данная книга. Вместе с тем, книга составлена так, что она может быть использована как практическое руководство для программистов, начинающих изучать ассемблер или желающих углубить свои знания о нем.

При подборе фактического материала автор стремился к тому, чтобы читатель получил достаточно полное представление об основных режимах SVGA, о том, как устроены графические задачи и какие вопросы приходится решать при их разработке. Насколько это ему удалось — судить вам, уважаемые читатели.

ГЛАВА 1



Видеосистемы и стандарт VESA

Персональный компьютер (далее ПК или РС) не был бы таковым при отсутствии внешних устройств. К ним относятся различные клавиатуры, "мыши", джойстики, принтеры, сканеры, модемы, звуковые карты, накопители на гибких, жестких, оптических и прочих дисках и, конечно же, мониторы. Пожалуй, наиболее важным из всех внешних устройств является оперативная память, поскольку без нее процессор просто не работоспособен. Вообще, внешним является любое устройство, не входящее в состав процессора (точнее микропроцессора).

Процессор не может непосредственно управлять работой внешнего устройства. Для этого нужен посредник — контроллер, который участвует в обмене данными между процессором и устройством и выполняет специфические действия, зависящие от особенностей устройства. Обычно контроллер обслуживает одно устройство. Исключением является контроллер ввода-вывода, обслуживающий все дисководы, а также порты параллельного и последовательного интерфейсов, к которым подключаются принтеры, мыши, джойстики и некоторые другие устройства.

Контроллеры могут располагаться на основной (системной, материнской) плате ПК, либо на отдельных платах (картах), вставляемых в разъемы основной платы. В отдельных случаях на карте может находиться и само внешнее устройство, например, внутренний модем.

Нас будут интересовать видеоконтроллеры, к которым подключаются мониторы. Большинство из них выполнено в виде отдельной платы, но в последние годы наметилась тенденция выпуска материнских плат с расположенными на ней ("интегрированными") видеоконтроллерами. Преимущество отдельных плат в том, что их всегда можно заменить другими с лучшими параметрами.

Для поддержки работы с любым контроллером требуется специальное программное обеспечение. Обычно оно записано на гибких или лазерных дис-

ках, прилагаемых к контроллеру, или входит в комплект операционной системы, например Windows 9X (95, 98, ME) и 2000. В любом случае при вводе нового устройства в эксплуатацию производится установка соответствующего программного обеспечения.

Данная глава содержит общий обзор современных компьютерных видеосистем и способов их программирования. Центральное место в ней занимает описание стандарта VESA, который оказал существенное влияние на развитие компьютерной графики.

1.1. Видеосистемы и их стандартизация

Компьютерный рынок динамичен по своей природе, он постоянно предлагает новое, все более совершенное видеоборудование для ПК. Пользователи, напротив, склонны соразмерять свои потребности и реальные финансовые возможности. Поэтому в эксплуатации находится множество видеосистем, изготовленных разными фирмами и в разное время. Они существенно различаются по конструктивному исполнению, техническим характеристикам и, конечно же, по особенностям программирования. На их примере можно проследить всю историю развития технических средств персональных компьютеров от первых IBM PC XT до новейших моделей на базе процессоров семейства Pentium.

1.1.1. Мониторы

Основными элементами компьютерных видеосистем являются мониторы и видеокарты. Кроме того, к ним относятся графические ускорители и платы для работы с телевизионным изображением, которые используются в зависимости от особенностей решаемых задач.

Монитор или дисплей является одним из важнейших узлов современного персонального компьютера, предназначенным для визуализации (т. е. для представления в виде, доступном для человеческого глаза) выводимой информации. За время существования ПК было выпущено достаточно много мониторов различных типов. По степени улучшения технических характеристик их можно расположить в такой последовательности: Color Graphics Adapter (CGA), Enhanced Graphics Adapter (EGA), Video Graphics Array (VGA), Super Video Graphics Array (SVGA). Этот перечень не претендует на полноту, но каждая из названных моделей получила, в свое время, наибольшее распространение. Мониторы CGA, EGA в настоящее время безнадежно устарели и не выпускаются промышленностью, однако некоторые из них еще находятся в эксплуатации.

Для получения изображения на экране монитора требуется выполнение специальных действий, поэтому неотъемлемой частью монитора является

встроенное в него устройство управления. Это еще не контроллер, речь идет о совершенно других функциях, поэтому в англоязычной литературе для названия таких устройств используется термин *unit*. Именно благодаря встроенному устройству управления все современные мониторы (VGA и SVGA) подключаются к стандартному разъему компьютера и поддерживают стандартный аппаратный интерфейс, т. е. определенную последовательность управляющих сигналов, их полярность, уровень и форму. Поэтому мониторы выпускаются и продаются как отдельные функциональные устройства без контроллера (так же, как, например, принтеры).

Для получения изображения в современных настольных мониторах используется электронно-лучевая трубка (кинескоп). Модели таких мониторов различаются по размеру экрана, разрешающей способности, используемому кинескопу, способам настройки (простые или с цифровым управлением). Эти характеристики определяют качество изображения и стоимость изделия, но никак не влияют на интерфейс с контроллером и способы программирования.

В переносных и портативных компьютерах применяются мониторы, в которых изображение получается на матрице из жидких кристаллов. На сегодняшний день качество такого изображения хуже, чем у мониторов с обычным кинескопом. Однако поиски новых принципов получения изображения на экране идут весьма активно и, возможно, в ближайшем будущем, будут достигнуты большие успехи в этом направлении.

Как любое внешнее устройство монитор не может быть подключен к ПК без специального контроллера, являющегося посредником между ним и центральным процессором (ЦП). Эти контроллеры легко отличить от других плат, поскольку на них расположен специальный разъем для подключения монитора. При описании такого типа контроллеров используют термины "видеоконтроллер", "видеокарта", "видеоплата", реже "видеоадаптер".

1.1.2. Видеокарты и стандарты

Видеокарты воспринимают цифровую информацию, поступающую от ЦП, и вырабатывают сигналы, управляющие работой монитора. Ядром видеокарты является специализированный микропроцессор, выполняющий все необходимые функции. От него зависят такие технические характеристики, как производительность (или быстродействие), предельно допустимый объем памяти, конкретные особенности программирования. Кроме того, на любой видеокarte расположена оперативная память (RAM), предназначенная для хранения цифрового образа, выводимого на экран изображения. Наконец, на видеокarte находится постоянная память (ROM), содержащая фрагмент базовой системы ввода-вывода (BIOS).

Системные шины. Конструктивное исполнение видеокарт соответствует определенным техническим стандартам. Каждая из них способна взаимодейст-

воват только с конкретным типом системной шины. Внешним признаком этого является форма разъема (гнезда) материнской платы, в которое устанавливается видеокарта. Системная шина расположена на материнской плате и представляет собой совокупность проводящих линий, по которым передаются данные, адреса и управляющие сигналы. От нее зависит такая важная характеристика, как скорость передачи данных, а следовательно, и время, затрачиваемое на построение изображения на экране.

На материнских платах компьютеров, собранных на базе процессоров Intel 80286 и Intel 80386 применялась шина ISA (Industry Standard Architecture), при использовании которой обмен данными между видеокартой и процессором производится словами или байтами. Для процессора Intel 486 была разработана новая системная шина VLB (VESA Local Bus), но ее очень скоро вытеснила шина PCI (Peripheral Component Interconnect). Обе шины позволяли передавать данные двойными словами. Большинство имеющихся в продаже современных видеокарт выполнено в стандарте PCI. С выпуском процессоров Pentium II на системной плате появилась специальная 128-рядная шина для обмена данными с видеокартой. Она заканчивается гнездом AGP (Accelerated Graphics Port). В настоящее время видеокарты, выполненные в стандарте AGP, преобладают на компьютерном рынке.

Для каждого типа шин выпускались и продолжают выпускаться не только видеокарты, но и платы различного назначения. Поэтому на материнских платах обычно имеются разъемы для установки карт, выполненных в стандартах ISA и PCI. Например, одна из современных материнских плат фирмы Intel (ее тип AL 440 LX) содержит 2 разъема ISA, 4 разъема PCI и 1 разъем AGP. Вполне возможно, что на современном ПК с процессором Pentium вы обнаружите видеокарту, выполненную в стандарте PCI и даже в устаревшем стандарте ISA.

Описанные различия видеокарт учитываются при их программировании лишь в особых случаях, когда требуется максимальная производительность видеосистемы. Большинство прикладных задач в этом не нуждается. Поэтому для нас более важно знать тип монитора, для обслуживания которого предназначена видеокарта, поскольку от этого зависят основные особенности ее программирования и структура графических задач.

Стандарты IBM. Свой первый персональный компьютер американская фирма IBM (International Business Machines) выпустила в 1981 году. В то время основным производителем персональных компьютеров была другая американская фирма DEC (Digital Equipment Corporation). IBM не входила даже в тройку лидеров, но за короткий отрезок времени она стала "законодателем мод" в сфере производства ПК. Этому, в немалой степени, способствовало то, что IBM публиковала подробную информацию о своих новых разработках, и ее могли использовать другие фирмы, занимающиеся производством компьютеров, совместимых с IBM PC, дополнительных плат различного назначения и разработкой программного обеспечения.

Впервые за всю историю существования IBM применила в своем изделии компоненты, изготовленные другими фирмами. В частности, в IBM PC использовался микропроцессор Intel 8086. С тех пор все семейство IBM PC базируется на микропроцессорах фирмы Intel. Кроме них могут применяться совместимые микропроцессоры фирм AMD и Cyrix.

Первая модель ПК выпускалась недолго, на смену ей пришел компьютер второго поколения PC XT, аббревиатура XT расшифровывается как eXtended Technology (расширенная технология). В нем по-прежнему использовался микропроцессор Intel 8086, но пространство оперативной памяти было увеличено до 640 Кбайт. Кроме того, были разработаны новые видеокарты, предназначенные для работы с монитором CGA и позволявшие отобразить 8 цветов. Объем видеопамати у них достигал 4 Кбайт, а разрешающая способность составляла 320×200 точек.

Следующая модель компьютера была создана на базе микропроцессора Intel 80286, она называлась PC AT, аббревиатура AT расшифровывается как Advanced Technology (прогрессивная технология). При разработке PC AT в качестве стандарта был принят монитор EGA. На видеокартах появился новый разъем для подключения монитора. Количество разных цветов точки возросло до 16-ти, а объем видеопамати до 64 Кбайт. Такой объем видеопамати и разрешающая способность монитора позволяли создавать на экране изображение размером 640×350 точек.

К этому времени производство видеокарт и мониторов освоили разные фирмы, и их продукция существенно различалась по техническим характеристикам. Выпускались карты, которые позволяли использовать 64 цвета и имели объем видеопамати больше, чем 64 Кбайт. Однако для использования таких карт требовалось описание способов их программирования, которое в большинстве случаев недоступно для программистов.

Для стандартов CGA и EGA характерна сложная организация видеопамати. Простая запись кода точки в видеопамать или чтение кода из нее невозможны, для этого требуется около десятка команд и приходится работать с портами видеокарты.

Стандарт на монитор VGA был опубликован при выпуске новой серии IBM PS (персональные системы) на базе процессора Intel 80386. Эта серия компьютеров не получила широкого распространения. Стандарт был принят, но в стремлении вырваться вперед IBM выпустила недоработанный продукт, чем и не замедлили воспользоваться конкуренты.

Стандарт VGA предусматривал новый трехрядный 15-контактный разъем для подключения монитора. Пожалуй, это наиболее продуманная часть стандарта. В разьеме были оставлены свободные (зарезервированные) контакты для будущих расширений. Поэтому его форма не изменилась до настоящего времени. При подключении современных SVGA-мониторов используются некоторые из ранее зарезервированных контактов разъема. Следует отметить,

что такой стандарт разъема распространяется только на видеокарты, предназначенные для семейства IBM PC.

Стандарт VGA был шагом вперед по количеству цветов, которое возросло с 16 до 256. Объем видеопамати увеличился до 256 Кбайт, и упростилась ее организация. В отличие от стандартов CGA и EGA, запись и чтение кода точки теперь производились одной командой, как при работе с обычной (оперативной) памятью компьютера. Однако стандарт VGA имел следующий существенный недостаток. Видеопамать, как и обычная память, делится на сегменты размером по 64 Кбайт. Стандарт VGA не предусматривал механизм переключения сегментов, поэтому на экране можно было отобразить содержимое только одного из них. Соответственно размер максимально возможного изображения составлял 320×200 точек ($320 \times 200 = 64\,000$, что немного меньше, чем 64 Кбайт).

Стандарты IBM и BIOS. Для программирования конкретной видеокарты надо знать назначение ее внутренних регистров, их размерность (байты, слова и пр.), способ записи или чтения данных и расположение величин в разрядах регистров. Первый стандарт IBM регламентировал назначение, состав и способы работы с внутренними регистрами, что исключало несовместимость видеокарт. Но скоро стало очевидно, что это плохой способ решения проблемы совместимости и в стандартах EGA и VGA указанные требования распространялись только на основную часть регистров.

Для решения проблемы совместимости были стандартизированы функции BIOS. IBM выпустила описание базового набора, содержащее перечень основных функций, способ их вызова, назначение и размещение входных и выходных параметров. Так появилась группа функций BIOS с названием "Video Services". Образующие ее подпрограммы и данные не входят в основную часть BIOS, они хранятся в специальной микросхеме, расположенной на видеокарте. Поэтому, устанавливая новую видеокарту, вы одновременно устанавливаете новую реализацию функций указанной группы. У современных моделей видеокарт эта группа может занимать полный сегмент памяти, т. е. 64 Кбайт. Это свидетельствует о сложности и разнообразии выполняемых действий и о большом объеме используемых при этом данных.

Именно благодаря наличию функций "Video Services" вы можете быть уверены в том, что после смены видеокарты ваш ПК сохранит свою работоспособность. Программисты получили существенное упрощение структуры прикладных задач и их независимость от моделей видеокарт. А разработчики получили возможность изменять программную реализацию функций BIOS для учета конкретных особенностей видеокарты.

Перед выводом на экран монитора текста или графических изображений должен быть установлен соответствующий видеорежим. В частности, при первоначальной загрузке ПК BIOS устанавливает текстовый режим работы, при котором на экране можно расположить 25 строк, каждая из которых

содержит не более чем 80 символов. DOS обычно не изменяет этот режим, а прикладные задачи могут выполняться в текстовых или в графических режимах.

В группу "Video Services" обязательно входит функция, выполняющая установку видеорежима. При ее вызове указывается код видеорежима, а данные, необходимые для его установки, хранятся в области BIOS. IBM ввела стандартные значения кодов для 20-ти видеорежимов, значения которых изменяются от 0 до 13h (буква h — признак шестнадцатеричного числа). Разработчики видеокарт могут вводить новые режимы по своему усмотрению, чем они обычно и пользуются.

Стандарты были не всегда. После неудачи со стандартом VGA IBM прекратила работы по стандартизации видеооборудования. А поскольку никто этим не занимался, то наступил период "разброда и шатаний". Каждая фирма проектировала платы по своему усмотрению, не заботясь о каком-либо общем стандарте, кроме собственного. В результате было выпущено много хороших, но не совместимых друг с другом видеокарт, поддерживающих видеорежимы с более высоким, по сравнению с VGA, геометрическим и цветовым разрешением. Коды и характеристики режимов существенно различались и программы, рассчитанные на работу с одной видеокартой, не могли работать с другими или, в лучшем случае, требовали дополнительной настройки. Разумеется, что программисты нашли выход и в системных библиотеках появились модули для определения типа установленной на компьютере видеокарты и настройки программы на ее параметры, но это были полумеры, требовалось радикальное решение.

При этом следует отметить, что отсутствие стандартов имело и свою положительную сторону. Именно в это время разработчиками видеооборудования был накоплен практический опыт использования различных видеорежимов. Трудно себе представить специалистов, которые могли бы предусмотреть все возможные случаи, не опираясь на существующий опыт. Намного проще обобщить достигнутые результаты, оставить главное и отбросить ненужное.

Необходимость стандартизации понимали не только программисты, но и производители видеооборудования. Благодаря объединению их усилий и была создана специализированная ассоциация VESA, которая до настоящего времени занимается вопросами стандартизации видеооборудования (не только для IBM PC). Все ведущие производители придерживаются этих стандартов и проблема несовместимости мониторов или видеокарт в наше время не столь актуальна, но технический прогресс вынуждает, время от времени, вновь возвращаться к проблеме стандартизации, уже на более высоком уровне работы с графикой.

Подведем итог сказанному в данном разделе. В современных видеокартах используется различная элементная база, поэтому они могут существенно

различаться по своим техническим характеристикам. Но при использовании функций BIOS все они без исключения совместимы на программном уровне с видеорежимами VGA IBM и VESA. Сказанное распространяется не только на обычные видеокарты, но и на акселераторы.

1.1.3. Акселераторы

В истории вычислительной техники развитие программных и аппаратных средств тесно переплетено друг с другом. Новые технические возможности позволяют программистам сделать очередной шаг вперед и при этом у них часто появляются новые требования к аппаратуре. Основное влияние на расширение функций видеокарт оказало стремление получить на экране монитора объемное движущееся изображение, построенное с учетом перспективы и распределения света и тени. Пионерами этого направления были разработчики компьютерных игр.

Трехмерная графика. Как известно, на холсте или листе бумаги можно нарисовать только плоское изображение, а для придания ему эффекта объемности применяются специальные приемы рисования. То, что мы видим в результате, является оптической иллюзией (обманом зрения), основанной на нашем жизненном опыте восприятия окружающего мира.

Для получения изображения куба его грани надо расположить под определенными углами. У шара граней нет, поэтому для придания кругу эффекта объемности используется распределение света и тени, создаваемое с помощью штриховки или раскрашивания. Для получения эффекта расположения в пространстве нескольких объектов их размеры уменьшаются по мере удаления от точки наблюдения.

Перечисленные приемы основаны не только на нашем субъективном восприятии окружающего мира, но и на вполне объективных законах оптики. Существует возможность формального описания способов преобразования трехмерных объектов в двухмерные и программной реализации алгоритмов вычислений. Такие алгоритмы не учитывают субъективный фактор, но это второстепенный вопрос.

В отличие от листа бумаги экран монитора является плоской дискретной поверхностью, поэтому компьютерная графика имеет дело с дискретными объектами. Для их аналитического описания нужны специальные методы аппроксимации. При описании плоских (двухмерных) объектов обычно применяются методы линейной и векторной графики, а при описании трехмерных объектов — полигональной графики (слово *polygon* переводится как многоугольник). Чаще всего в качестве многоугольников используются треугольники.

В разработку методов трехмерной графики основной вклад внесла Reality Lab 3D, подразделение ныне не существующей компании Rendermorphis.

Результаты ее работы легли в основу графической библиотеки Direct3D, продукт фирмы Microsoft. Они же были использованы при создании языка VRML, предназначенного для описания трехмерных сценариев. Его создала Cosmo Software, а адаптировала российская фирма ParaGraph.

Процесс построения трехмерного изображения можно разделить на два этапа:

- геометрические вычисления;
- визуализация полученных результатов.

Геометрические вычисления сводятся к манипуляциям с векторами и матрицами, в результате которых получается совокупность треугольников, аппроксимирующая поверхность трехмерного объекта. Эти вычисления полностью зависят от конкретных свойств графических объектов, поэтому их выполняет центральный процессор ПК.

Процесс визуализации (rendering) заключается в том, что полученные треугольники отображаются на дискретную плоскость (поверхность экрана). При этом некоторые из них могут превратиться в линии или точки, а часть окажется на невидимой стороне объекта. Треугольники раскрашиваются по заданным образцам (их называют текстурами), а для получения эффекта объемности учитывается распределение уровней освещенности. Кроме того, могут понадобиться вычисления, связанные с изменением размеров всего изображения или отдельных его частей, коррекцией перспективы, созданием эффекта тумана и др.

Все эти действия не столь сложны, как геометрические построения, но их количество огромно, оно во много раз больше, чем количество точек на экране монитора или в рисунке, если он занимает не весь экран. Выполнение визуализации процессором ПК существенно замедляет работу с графикой.

Одним из распространенных приемов ускорения является перенос несложных, но многократно повторяемых вычислений на аппаратный уровень. Для этого создаются специальные процессоры, выполняющие нужные вычисления по микропрограммам, работающим намного быстрее программ аналогичного назначения.

Модели акселераторов. Акселератор (accelerator, ускоритель) является специализированным вычислительным устройством, предназначенным для ускорения процесса построения или преобразования графических изображений. В отличие от обычных видеокарт акселератор получает от процессора не законченный образ изображения, а более общую и сжатую информацию, на основании которой он, а не процессор, вычисляет точечный образ рисунка и записывает его в видеопамять.

Первые модели акселераторов выпускались в виде отдельных плат, которые обрабатывали данные, поступающие от процессора, и передавали их обычной видеокарте. Но очень скоро вычислительный микропроцессор начали

располагать на плате видеокарты, а затем его и видеоконтроллер объединили в одну большую микросхему (чип). Акселераторы могут предназначаться для ускорения работы с двухмерными или трехмерными графическими объектами. В названии первых присутствует обозначение 2D, а в названии вторых — 3D, где D является первой буквой слова *direction* — направление.

Акселераторы могут быть рассчитаны на установку в гнездо шины PCI или в специализированный разъем AGP, который появился на материнских платах ПК после выпуска микропроцессора Pentium II. Разъем AGP имеет 128-рядную шину данных, что существенно ускоряет процесс обмена между центральным процессором и видеокартой.

При вычислениях используется видеопамять, расположенная на плате акселератора, поэтому ее объем всегда больше того, который нужен для работы видеокарты в обычном режиме. В настоящее время в продаже имеются акселераторы AGP с объемом видеопамати 32 и 64 Мбайт.

Современные модели акселераторов чаще всего собираются на базе специализированного графического процессора, выполненного в виде большой интегральной микросхемы (чипа), которые выпускают более десятка различных фирм. Корпорация Intel выпустила свой чип i740, но пока он не получил широкого распространения. По данным агентства Mercury Research на сентябрь 1998 года, в первую пятерку производителей графических чипов входят: ATI, S3, Cirrus Logis, Silicon Integrated System (SIS) и Trident Microsystems. Кроме основных чипов эти фирмы выпускают комплекты сопутствующих микросхем и полностью завершённую продукцию, т. е. видеокарты и акселераторы. Возглавляющая список ATI Technologies производит 27% всех графических чипов. Поэтому вероятность приобрести акселератор, собранный из ее комплектующих, весьма велика.

Акселераторы существенно различаются по цене, но возможности дешевых моделей ограничены. Обычно у них разрешающая способность не превышает 800×600 точек, а из полноцветных видеорежимов поддерживается только Hi-Color. Однако при использовании акселератора в качестве обычной видеокарты эти ограничения отсутствуют.

Функции акселераторов. Набор выполняемых функций зависит от конкретного назначения акселератора. По личному опыту вы знаете, что при рисовании и черчении приходится иметь дело с различными объектами и с разными способами их изображения. Соответственно, функции акселераторов, предназначенных для ускорения геометрических построений и для работы с художественными изображениями, различаются весьма существенно. Некую комбинацию из этих функций могут поддерживать акселераторы, предназначенные для систем автоматизации проектирования. Специфический набор функций поддерживают акселераторы, используемые для игровых приложений. В этих случаях основными требованиями являются быстрота

смены картинки и возможность создания различных спецэффектов, а точность построения самого изображения не столь существенна.

На сегодняшний день невозможно выделить некий стандартный набор функций, выполняемых акселераторами. У разработчиков отсутствует достаточный практический опыт, поэтому они просто воплощают функции, поддерживаемые графическими библиотеками Direct3D и OpenGL. К ним относятся раскрашивание треугольников по заданным образцам (наложение текстур), альфа смешение (см. раздел 7.6), создание эффекта тумана (см. раздел 7.6), вычисление распределения света и тени по методу Гуро (Gouraud Shading), коррекция перспективы и некоторые другие.

Кроме того, у акселераторов появилась функция иного назначения. Это преобразование телевизионного изображения в компьютерное и обратно. В таких случаях на видеокarte имеется разъем для подключения телевизора или видеомагнитофона. В качестве примера можно привести изделия фирм ATI и S3. Подчеркнем, что речь идет не о приеме телевизионных сигналов — для этого существуют специальные платы, а об аппаратном преобразовании сигналов из телевизионного стандарта (NTSC, PAL и т. п.) в последовательность кодов точек, записываемых в видеопамять. Такое преобразование является двухсторонним, т. е. коды хранящихся в видеопамети точек могут преобразовываться в один из телевизионных стандартов. Такие функции расширяют возможности компьютерной обработки телевизионных изображений и делают ее более доступной для пользователей.

Программирование акселераторов. После включения ПК и загрузки операционной системы, неважно какой, акселератор работает как обычная видеокarta. Такой режим необходим для нормальной работы операционных систем и многих прикладных задач. Примером могут служить Windows и ее многочисленные приложения. Программирование акселератора как обычной видеокарты ничем не отличается от того, что описано в данной книге.

Основные (вычислительные) функции акселераторов выполняются в 32-разрядном (защищенном) режиме работы ПК. Речь идет о разрядности адресов, данные могут содержать меньшее или большее количество разрядов. Выполнение прикладных задач в защищенном режиме поддерживают, например, Windows 9X/2000/NT и OS/2. DOS является операционной системой реального (16-разрядного) режима, но существуют так называемые расширители (DOS extenders), которые подключаются к прикладной задаче и создают на время ее выполнения вычислительную среду, необходимую для работы в защищенном режиме. Наиболее известными из них являются DOS4GW, DOS32A, PMODE/W.

При выборе операционной системы приходится учитывать тот факт, что в настоящее время не существует стандарта на программирование функций акселераторов, хотя ассоциация VESA предпринимает активные усилия по его разработке. Первый документ VBE/AF Standard 1.0 был выпущен в августе 1996 года. В настоящее время опубликована третья редакция этого доку-

мента, но в ее первых строках подчеркивается, что она содержит черновые предложения и не более того.

Отсутствие стандартов означает, что непосредственно взаимодействующая с акселератором задача не будет переносимой. Она будет выполняться только на тех ПК, на которых установлена соответствующая модель акселератора. Существует довольно много компьютерных игр, созданных для определенных моделей акселераторов, в остальных случаях они либо вообще не работают, либо работают медленно, если вычисления выполняет процессор.

Проблема переносимости частично решается с помощью драйверов, которые продаются вместе с акселератором. Они составлены для определенной операционной системы и рассчитаны на взаимодействие с одной из распространенных графических библиотек.

Среда Windows позволяет создавать любые графические приложения. Разработчикам доступны графические библиотеки Direct3D и OpenGL, которые хорошо документированы и общедоступны. Если установлен соответствующий драйвер, то они используют возможности акселератора, в противном случае требуемые действия выполняются программно, что замедляет процесс выполнения задач, но решает проблему их переносимости.

Пакет Direct3D разработан Microsoft и является одной из частей библиотеки DirectX, входящей в состав Windows 9X, начиная с версии 98, Windows NT и 2000. Он предназначен для ускорения выполнения игровых задач в среде Windows. Первая версия пакета была выпущена в 1996 году.

Библиотека OpenGL была создана в 1993 году фирмой Silicon Graphics для компьютеров совершенно другого класса и для иной операционной системы. В 1995 году совместно с Microsoft она адаптировала ее для IBM PC. С этой библиотекой работает, например, последняя версия компилятора Фортрана для Windows.

В отличие от Direct3D, OpenGL более гибкая и многофункциональная библиотека. Изначально она создавалась для применения трехмерной графики в системах автоматизированного проектирования. Однако она не содержит средств, позволяющих работать непосредственно с видеопамятью в обход интерфейса графических устройств (GDI), который существенно замедляет выполнение прикладных задач. Для этого нужна дополнительная библиотека WinG или DirectDraw, которая является частью библиотеки DirectX.

Еще совсем недавно мысль о том, что можно создать хорошее трехмерное графическое приложение под Windows казалась совершенно нелепой любому программисту, имеющему дело с этой системой. С появлением библиотеки DirectX ситуация изменилась в лучшую сторону. Но, тем не менее, среда Windows остается весьма инерционной, и разработчики компьютерных игр продолжают и, вероятно, еще долго будут продолжать использовать для ускорения процесса игры расширитель DOS4GW и ему подобные.

1.2. Общая характеристика стандарта VESA

Video Electronics Standards Association (ассоциация стандартизации видеоэлектроники), сокращенно VESA, была основана в 1989 году. В августе того же года она опубликовала свой первый стандарт для 16-цветного видеорежима SVGA с разрешением 800×600 точек. С тех пор ассоциация выпустила множество различных стандартов, охватывающих широкий спектр видеоборудования. Одной из ее известных разработок является стандарт на системную шину VLB (VESA Local Bus) для микропроцессора Intel 486. Однако, как уже говорилось, эта шина не прижилась.

Если у вас есть доступ к сети Internet, то подробные сведения об ассоциации VESA и ее продукции можно найти на серверах www.vesa.org и [ftp.vesa.org](ftp://vesa.org).

Нас будут интересовать стандарты VESA, регламентирующие способы программирования видеокарт. Первый законченный стандарт появился в октябре 1991 года, он определял полный набор видеорежимов SVGA и дополнительных функций BIOS и назывался VESA BIOS Extension (VBE) version 1.2. Это функции той части BIOS, которая расположена на видеокартах и обслуживает видеосервис. Стандарт объединил предыдущие версии VBE 1.0 и VBE 1.1. Ему соответствуют практически все видеокарты, изготовленные начиная с 1992 года. Более современные видеокарты поддерживают версию VBE 2.0, которая совместима (сверху вниз) с версией VBE 1.2. Поэтому учет рекомендаций VESA при программировании работы с графикой позволяет создавать переносимые задачи, которые будут правильно работать независимо от модели видеокарты, установленной на конкретном компьютере.

1.2.1. Стандартизация видеорежимов

Понятие "видеорежим" является обобщенной характеристикой текущего состояния видеоконтроллера. Основная функция видеоконтроллера состоит в отображении содержимого видеопамати на экране монитора. Выполнение этой функции зависит от множества величин, хранящихся во внутренних регистрах видеоконтроллера. Значения этих величин определяются при установке видеорежима. Нас интересуют те из них, которые не только влияют на работу видеокарты, но и должны учитываться в прикладных задачах.

Характеристики видеорежимов. Прежде всего, видеорежимы делятся на *текстовые* и *графические*. В зависимости от типа режима прикладная задача записывает в видеопамать или коды символов в стандарте ASCII, или коды отдельных точек графического объекта. При работе в графических режимах видеоконтроллер просто выводит на экран точки, коды которых хранятся в видеопамати. При работе в текстовых режимах он, по кодам символов,

выбирает их изображения из специальных таблиц, а затем выводит точки изображений на экран.

Другой важной характеристикой является *разрешающая способность*. В зависимости от типа видеорежима она измеряется количеством символов или точек, которое можно разместить по горизонтали и вертикали в пределах рабочей области экрана. Количество точек является основной, а количество символов — производной единицей, т. к. оно зависит от первой величины и от размеров ячейки (знакоместа), отведенной для размещения одного символа.

Точки, расположенные по горизонтали, образуют строку, а по вертикали — столбец (в документации на BIOS используются термины *row* (ряд) и *column* (столбец)). Количество точек в строке и в столбце не может быть произвольным, оно всегда кратно восьми. Максимально возможное количество точек в строке зависит от разрешающей способности монитора и его геометрических размеров. У современных мониторов минимальное расстояние между центрами смежных точек составляет от 0,28 до 0,26 мм. При размере экрана 14 дюймов по диагонали количество точек в строке не превышает 1024. У 15-дюймовых мониторов оно достигает значения 1280. Однако возможность работы в режимах с высоким разрешением зависит еще и от видеокарты, о чем будет сказано ниже.

Расстояние между соседними точками, расположенными по горизонтали и вертикали, подбирается одинаковым, для того чтобы изображение квадрата на экране выглядело как квадрат, а не как прямоугольник. Обычно количество точек по горизонтали больше, чем по вертикали, но существуют мониторы и с вертикальной ориентацией страницы.

Рабочая область никогда не заполняет всю видимую часть экрана. Во всех видеорежимах ее окружает пространство, которое в документации называется *overscan* или *border* (граница, кайма). Поэтому в разных режимах геометрические размеры рабочей области могут не совпадать.

Важной характеристикой видеорежимов является *количество цветов*, которое можно одновременно отобразить на экране. Во всех графических режимах цвет получается в результате совмещения в одной точке экрана трех базовых цветов (красного, зеленого и синего) разной интенсивности. В зависимости от видеорежима коды базовых цветов располагаются либо в специальных регистрах видеокарты, либо в видеопамати, т. е. непосредственно в коде точки. Первую категорию режимов принято называть *packed pixel graphics* (упакованная точечная графика), а вторую — *direct color* (непосредственный цвет). Вторая категория, в свою очередь, делится на режимы *Hi-Color* и *True Color*. В любом случае от видеорежима зависят размер кода точки и размеры кодов базовых цветов.

Видеорежимы VESA. Разработчикам стандарта VESA предстояло, в первую очередь, ограничить разнообразие применявшихся на практике видеорежи-

мов, связав с каждым из них конкретный код и набор характеристик. В двух первых версиях стандарта было описано 8 графических режимов `packed pixel graphics` и 5 текстовых режимов высокого разрешения. Графические видеорежимы `direct color` были введены в третьей версии стандарта. Коды режимов VESA и их характеристики перечислены в табл. 1.1.

Таблица 1.1. Видеорежимы VESA

Код режима	Количество точек в строке	Количество строк по вертикали	Размер точки в битах	Размер строки в байтах	Количество цветов
VESA редакции 1.0 и 1.1					
100h	640	400	8	640	256
101h	640	480	8	640	256
102h	800	600	4	100	16
103h	800	600	8	800	256
104h	1024	768	4	128	16
105h	1024	768	8	1024	256
106h	1280	1024	4	160	16
107h	1280	1024	8	1280	256
108h	80	60	—	160	16
109h	132	25	—	264	16
10Ah	132	43	—	264	16
10Bh	132	50	—	264	16
10Ch	132	60	—	264	16
VESA редакции 1.2					
10Dh	320	200	15	640	32K
10Eh	320	200	16	640	64K
10Fh	320	200	32/24	1280/960	16M
110h	640	480	15	1280	32K
111h	640	480	16	1280	64K
112h	640	480	32/24	2560/1920	16M
113h	800	600	15	1600	32K
114h	800	600	16	1600	64K

Таблица 1.1 (окончание)

Код режима	Количество точек в строке	Количество строк по вертикали	Размер точки в битах	Размер строки в байтах	Количество цветов
VESA редакции 1.2 (прод.)					
115h	800	600	32/24	3200/2400	16M
116h	1024	768	15	2048	32K
117h	1024	768	16	2048	64K
118h	1024	768	32/24	4096/3072	16M
119h	1280	1024	15	2560	32K
11Ah	1280	1024	16	2560	64K
11Bh	1280	1024	32/24	5120/3840	16M

В первом столбце табл. 1.1 перечислены коды видеорежимов. Это шестнадцатеричные числа, поэтому в их записи могут встречаться не только цифры, но и латинские буквы от А до F, а в конце кода обязательно указывается латинская буква h (100h = 256). В остальных столбцах таблицы приведены десятичные числа. В последнем столбце, для сокращения записи количества цветов, использованы буквы K и M. Они обозначают степени числа два, наиболее близкие по значению к тысяче (K=1024) и миллиону (M=1048576). Соответственно, количество цветов может быть следующим: 32K = 32768, 64K = 65536, 16M = 16777216.

В графических режимах размер строки указан в точках, а в текстовых — в виде количества символов. В текстовых режимах ширина символов постоянна и составляет 8 точек, а высота — 8 или 16 точек.

Классификация режимов. Перечисленные в табл. 1.1 видеорежимы делятся на следующие пять групп:

- текстовые режимы;
- 16-цветные режимы (EGA);
- packed pixel graphics (256 цветов);
- Hi-Color (direct color 32K или 64K цветов);
- True Color (direct color 16M цветов).

Этим группам соответствуют 4 разные модели видеопамати, поскольку разновидности режимов direct color используют одну модель. В описании стандарта понятие "модель видеопамати" четко не объясняется, но речь идет

о том, как видеоконтроллер интерпретирует содержимое байтов видеопамя-ти. Для программиста важно знать не модель, а способ доступа к видеопам-яти и что при этом записывается в ее байты.

Три из четырех моделей допускают непосредственную работу с видеопам-ятью, т. е. запись и чтение содержимого ее байтов и слов с помощью обычных команд ассемблера. Исключением являются режимы EGA, в этом случае для чтения или записи необходима работа с внутренними регистрами видеокарты. Эти режимы морально устарели, кроме того, они хорошо опи-саны в литературе, поэтому в данной книге не рассматриваются.

Текстовые режимы VESA просто расширяют возможности аналогичных ре-жимов IBM и позволяют использовать стандартные процедуры BIOS, предна-значенные для работы с текстом. Программирование в текстовых режимах описано в первой части главы 5.

Режимы *packed pixel graphics* отличаются от режима VGA IBM тем, что введено сегментирование видеопамя-ти, все пространство которой делится на окна размером по 64 Кбайт. Своевременное переключение окон позволяет работать с большим пространством видеопамя-ти, которое требуется для поддержки видеорежимов с высоким разрешением. Напомним, что в режи-ме VGA IBM разрешение составляет 320×200 точек (сравните с табл. 1.1).

В режимах *packed pixel graphics* между кодами точки и цвета нет одно-значного соответствия, поскольку они расположены в разных устройствах видеокарты. Коды точек хранятся в видеопамя-ти, а коды их цветов — в спе-циальных регистрах видеокарты. Изменяя содержимое этих регистров, мож-но изменить все цвета, использованные в изображении, без обращения к видеопамя-ти, т. е. не изменяя кодов точек образа рисунка. Количество реги-стров (256) определяет размер кодов точек — 1 байт (8 разрядов). В главе 3 описано программирование рисования и построения графических объектов в этих режимах, а в главе 4 — работа с цветом.

В режимах *direct color* базовые цвета расположены непосредственно в коде точки, который может содержать 2, 3 или 4 байта. На момент написания данной книги трехбайтовый код был обнаружен только у одного семейства акселераторов фирмы ATI. Размещение базовых цветов в коде точки значи-тельно расширяет возможности работы с цветом и позволяет создавать раз-личные спецэффекты, которые широко распространены в современной гра-фике. Программирование в режимах *direct color* описано в главе 7. В этих режимах увеличивается размер кода точки, а следовательно, и пространство видеопамя-ти, необходимое для хранения содержимого рабочей области эк-рана. Например, 1 Мбайт видеопамя-ти достаточно для работы во всех режи-мах *packed pixel graphics*, кроме 107h, с разрешением 1280×1024 точки. В то время как для поддержки режима 112h с разрешением всего 640×480 то-чек требуется 2 Мбайт, а для режима 11Bh с разрешением 1280×1024 необхо-димо 6 Мбайт видеопамя-ти. Поэтому возможность использования режимов

с высоким разрешением зависит не только от монитора, но и от объема видеопамяти.

Выбор конкретного режима зависит от программиста и от особенностей задачи, которую ему предстоит решать. В режимах `packed pixel graphics` достигается максимальная производительность видеосистемы, но ограничены возможности манипуляций с цветом. В режимах `direct color` увеличиваются затраты оперативной и видеопамяти и замедляется процесс построения графических объектов, но существенно расширяются возможности работы с цветом. Вариантов много, есть из чего выбирать.

Коды режимов VESA и OEM. Код используется не только при установке видеорежима, но и во многих других случаях. Поэтому, со времен IBM, он хранится в специальном байте оперативной памяти, расположенном в области данных BIOS с абсолютным адресом `0449h`. Старший разряд этого байта имеет специальное назначение, поэтому для записи кода видеорежима остается 7 разрядов и он может изменяться от 0 до `7Fh`. Первые 20 значений (от 0 до `13h`) отведены для кодов режимов IBM. Использование остальных значений кодов ничем не регламентировано.

Значения кодов видеорежимов, соответствующих стандарту VESA, изменяются от `100h` до `11Bh`. Такие числа не могут быть записаны в байт и, тем более, в его семь разрядов. Поэтому разработчики видеокарт по своему усмотрению заменяют 9-разрядные коды VESA 7-разрядными кодами OEM. Original Equipment Manufacturer (OEM) в дословном переводе означает "изготовитель оригинального оборудования", в нашем случае — изготовитель видеокарты, которая может собираться из микросхем и деталей других фирм. После установки режимов VESA в байте с адресом `0449h` хранятся коды OEM, а в состав BIOS входит специальная таблица их соответствия режимам VESA. Коды OEM уникальны для каждой модели и могут не совпадать даже у видеокарт одного семейства.

По своему усмотрению разработчики видеокарт могут вводить дополнительные режимы, отличающиеся по характеристикам от режимов VESA и IBM. Например, у акселераторов фирм ATI и S3 добавлены режимы с разрешением `320×400`, `400×300` и `512×384` точек. Можно предположить, что их удобно использовать при работе с кадрами телевизионных изображений. В соответствии с требованиями VESA коды и характеристики дополнительных режимов должны быть указаны в информационных блоках, хранящихся в области BIOS. Структура этих блоков и способ доступа к ним описаны в следующем разделе данной главы.

Набор режимов, введенный в VBE 1.2, VESA больше никогда не изменяла. В настоящее время выпускаются 20-дюймовые мониторы и видеокарты с объемом памяти 4 Мбайт и более. Это позволяет вводить новые видеорежимы с разрешением `1600×1200` точек. Такие видеорежимы поддерживают, например, видеокарты фирм S3 (VIRGE) и Matrox, в обоих случаях объем видеопамя-

ти составляет 4 Мбайт. При этом код режима `packed pixel graphics` с разрешением 1600×1200 точек в одном случае `120h`, а в другом — `11Ch`. Кроме того, карты Matrox поддерживают режимы `hi-color` с указанным разрешением.

Из всего сказанного следует, что работу с новой для вас видеокартой надо начинать с получения исчерпывающей информации о кодах поддерживаемых видеорежимов и их характеристиках. Для этого имеет смысл составить простую программу, которая будет выводить на экран (а лучше в файл) все нужные данные. Способ определения этих данных не сложен, он описан в следующем разделе данной главы.

1.2.2. Информационные функции VBE

Авторы стандарта VESA стремились не только облегчить работу программистов, но и не ограничивать разработчиков в выборе способов улучшения характеристик видеокарт. Компромиссным решением было включение в состав BIOS специальных информационных блоков с основными данными о видеокarte. В частности, они содержат список и характеристики всех поддерживаемых видеорежимов. В данном разделе описана структура основных информационных блоков и способ доступа к ним прикладных задач.

Вызов функций VBE. На любой видеокarte имеется микросхема пассивной, т. е. доступной только для чтения (ROM), памяти, в которой хранится фрагмент BIOS, содержащий структуры данных и подпрограммы, предназначенные для поддержки работы видеосистемы. В частности, к ним относятся функции, обращение к которым (вызов которых) происходит через прерывание `int 10h` (Video Services).

В состав группы Video Services обязательно входит набор функций для поддержки стандартных режимов IBM. Он необходим, по крайней мере, для нормального выполнения процесса загрузки ПК. Дополнением к нему являются функции VBE, описанные в данном и двух следующих разделах.

Перед обращением к BIOS код запрашиваемой функции помещается в регистр `ax`. Он состоит из кодов группы и функции в группе. Код группы VBE равен `4Fh`, он указывается в старшем байте регистра `ax`. Код функции для версии VBE 1.2 может изменяться от 0 до 8, он указывается в младшем байте регистра `ax`. Таким образом, содержимое регистра `ax` при вызове функций VBE 1.2 может изменяться от `4F00h` до `4F08h`.

Функции могут иметь входные и выходные параметры, которые передаются в регистрах общего назначения или в сегментных регистрах. Входные параметры нужны для нормального выполнения конкретной функции, а выходные содержат ее возвращаемый результат. Назначение и размещение параметров в регистрах будет описано для каждой функции.

Если запрошенная задачей функция поддерживается BIOS, то в регистр `al` возвращается код `4Fh`.

Важно

Это признак того, что функция могла быть выполнена. При успешном выполнении дополнительно очищается байт `ah`. В противном случае он содержит код ошибки. Таким образом, код `4Fh` в регистре `ah` является признаком успешного выполнения запроса.

Примеры вызова информационных функций `VBE` и использования возвращаемых ими данных описаны в главе 2.

Запрос общих данных. Для получения общих данных о видеокарте предназначена функция `4F00h Get SuperVGA Information`. Входным параметром является адрес массива размером 256 байтов, при исполнении запроса в него записываются данные о видеокарте. Полный адрес этого массива указывается в регистрах `es:di`. Форма записи `es:di` общепринята, она означает, что в регистре `es` находится сегмент памяти, а в регистре `di` — расположение (смещение) массива в этом сегменте. При исполнении запроса только первые 20 байтов массива заполняются следующими данными:

- 00 4 байта — `VESASignature`;
- 04 2 байта — `VESAVersion`;
- 06 4 байта — `OEMStringPtr`;
- 0Ah 4 байта — `Capabilities`;
- 0Eh 4 байта — `VideoModePtr`;
- 12h 2 байта — `TotalMemory`.

В первом столбце приведенного списка указаны смещения полей относительно начала массива, адрес которого находится в регистрах `es:di`.

Поле `VESASignature` содержит ASCII-коды четырех букв, образующих слово "VESA". Вот эти коды — `56h`, `45h`, `53h`, `41h`.

Поле `VESAVersion` занимает 2 байта, содержащих номер версии и ее редакцию, например `0102` для `VBE 1.2` или `0200` для `VBE 2.0`.

В поле `OEMStringPtr` находится полный адрес (из области `BIOS`) начала строки текста, содержащей наименование изготовителя видеокарты. Коды символов соответствуют стандарту ASCII, а строка заканчивается пустым байтом (формат строки `ASCIIZ`). Полный адрес занимает два слова, в первом из них хранится смещение, а во втором — код сегмента памяти.

Поле `Capabilities` состоит из 32-х независимых разрядов (битов), в которых указываются специфические особенности видеокарты. Авторы стандарта явно перестарались, даже в новейшей версии `VBE 3.0` описано назначение только пяти младших разрядов.

У рядовой видеокарты все 32 разряда поля `Capabilities` очищены.

Установка нулевого разряда означает возможность увеличения количества разрядов регистров `DAC` до 8 (см. описание функции `4F08h`).

Установка первого разряда означает, что видеоконтроллер не совместим с режимом IBM VGA.

Установка второго разряда означает необходимость синхронизации момента изменения содержимого регистров DAC с обратным ходом луча (см. описание функции 4F09h).

Третий и четвертый разряды описаны только в VBE 3.0, они устанавливаются в тех случаях, когда видеоконтроллер или внешние устройства поддерживают работу со стереоскопическими сигналами.

В поле VideoModePtr находится адрес начала списка видеорежимов, поддерживаемых картой. Первое слово поля содержит смещение, а второе сегмент. Список расположен в области BIOS, код каждого режима занимает одно слово. Признаком конца списка является код 0FFFFh.

В поле TotalMemory указан установленный на видеокарте объем памяти, выраженный в блоках размером 64 Кбайт. 1 Мбайт соответствует 16 блокам (10h). На устаревших моделях видеокарт это поле может быть очищено.

В каких случаях полезна описанная функция? Например, если графическая задача рассчитана на выполнение в защищенном (32-разрядном) режиме работы микропроцессора, то обязательно надо проверять номер версии VBE. Видеокарта может работать в таком режиме, если на ней реализованы функции VBE 2.0, но не VBE 1.2.

Анализ состояния разрядов поля Capabilities и списка поддерживаемых режимов при выполнении задачи едва ли целесообразен. Поддержка выбранного режима обязательно проверяется в задаче, но делается это более надежным способом, чем просмотр списка, поскольку присутствие режима в списке еще не означает его поддержку. Подробно это обсуждается в главе 2.

Целесообразно составить простую программу, которая сохраняет в файле или распечатывает результаты выполнения запроса 4F00h, включая список видеорежимов, и учитывать эти результаты при программировании.

Запрос характеристик видеорежимов. В начале выполнения любой графической задачи обязательно вызывается другая информационная функция, которая возвращает данные, необходимые для настройки на работу с конкретной видеокартой. Здесь описана структура информационного блока, а процессу настройки посвящена специальная глава 2.

Функция 4F01h Get SuperVGA Mode Information позволяет получить информацию о любом из поддерживаемых видеорежимов независимо от того, установлен он или нет. Ее целесообразно вызывать до попытки установить режим, т. к. полученные данные позволяют определить, поддерживает видеокарта работу в нужном режиме или нет, и выполнить ряд подготовительных действий.

Перед вызовом в регистрах es:di указывается адрес массива размером в 256 байтов (как и для функции 4F00h), а в регистр cx помещается код инте-

ресующего вас режима. Если видеокарта содержит VBE, то при возврате в регистре `ax` записан код `4Fh`. Если режим не поддерживается, то признак ошибки не вырабатывается, просто очищаются все поля табл. 1.2.

Результатом исполнения запроса является структура данных, приведенная в табл. 1.2. В первом столбце таблицы указаны смещения полей от начала массива, адрес которого находится в регистрах `es:di`. Второй столбец содержит размеры полей в байтах. Для примера в трех последних столбцах показаны значения величин, формируемых видеокартой `VIRGE /DX /GS` семейства `S3` при запросе характеристик режимов `101h` (640×480, 256 цветов), `110h` (640×480, 32К) и `112h` (640×480, 16М цветов). В шести случаях последние столбцы слиты в один это означает, что данные в соответствующих полях зависят не от режима, а от характеристик видеокарты.

Таблица 1.2. Информация, возвращаемая по запросу `4F01h`

Адрес поля	Размер поля	Что хранится в поле	Режим 101h	Режим 110h	Режим 112h
00	2	Атрибуты режима	009Bh	009Bh	009Bh
02	1	Атрибуты окна А	07		
03	1	Атрибуты окна В	00		
04	2	Размер ячейки окна в Кбайт	0040h		
06	2	Размер окна в Кбайт	0040h		
08	2	Код видеосегмента окна А	A000h	A000h	A000h
0Ah	2	Код видеосегмента окна В	A000h	A000h	A000h
0Ch	4	Адрес подпрограммы BIOS	556Ch C000h		
10h	2	Размер строки в байтах	0280h	0500h	0A00h
12h	2	Размер строки в точках	0280h	0280h	0280h
14h	2	Количество строк на экране	01E0h	01E0h	01E0h
16h	1	Ширина символа (текст)	08	08	08
17h	1	Высота символа (текст)	10h	10h	10h
18h	1	Количество планов памяти	01	01	01
19h	1	Количество бит на точку	08	0Fh	20h
1Ah	1	Количество банков видеопамати	01	01	01
1Bh	1	Модель видеопамати	04	06	06
1Ch	1	Размер банка в Кбайт	00	00	00