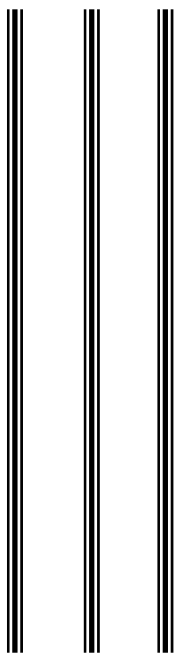


**ПРОГРАММИРОВАНИЕ
ВИДЕОСИСТЕМ**



• ИЗДАТЕЛЬСТВО ТГТУ •

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ТАМБОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**ПРОГРАММИРОВАНИЕ
ВИДЕОСИСТЕМ**

Методические указания к лабораторным работам для студентов 3 курса
специальности 2203 дневного отделения

Тамбов
• Издательство ТГТУ •
2003

УДК 681.3(075)
ББК \mathcal{Z} 973.26-018.2 я 73-5
П78

Утверждено Редакционно-издательским советом университета

Рецензент
Доктор технических наук, профессор
А. А. Чуриков

Составитель
С. А. Васильев

П78 Программирование видеосистем: Метод. указ. / Сост. С. А. Васильев. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2003. 24 с.

Методические указания содержат описание схем программирования VGA и Super VGA (в режиме VESA 1.2). Представлено описание основных программируемых регистров видеокарт. Рассмотрены основные режимы работы VGA. Представлены справочные данные по функциям VESA 1.2.

Предназначены для студентов 3 курса специальности 2203 дневного отделения.

УДК 681.3(075)
ББК \mathcal{Z} 973.26-018.2 я 73-5

© Тамбовский государственный
технический университет
(ТГТУ), 2003

УЧЕБНОЕ ИЗДАНИЕ

ПРОГРАММИРОВАНИЕ
ВИДЕОСИСТЕМ
МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Составитель ВАСИЛЬЕВ Сергей Александрович

Редактор Е. С. Мордасова
Инженер по компьютерному макетированию Т. А. Сынкова

Подписано к печати 26.02.2003.

Формат 60 × 84/16. Гарнитура Times. Бумага газетная. Печать офсетная.

Объем: 1,4 усл. печ. л.; 1,3 уч.-изд. л.

Тираж 75 экз. С. 135

Издательско-полиграфический центр
Тамбовского государственного технического университета
392000, Тамбов, ул. Советская, 106, к. 14

Видеокарты VGA и Super VGA

Микрокомпьютерные видеосистемы продолжают развиваться. Со времени появления персонального компьютера IBM PC в 1981 г. возможности графических адаптеров расширились. В историю ушли такие видеостандарты, как CGA, EGA. На сегодняшний день наибольшее распространение получили видеокарты VGA и SuperVGA. В данном комплексе лабораторных работ рассмотрены основные режимы чтения и записи пикселей видеоадаптеров, работающих в стандарте VGA и Super VGA (VESA 1.2).

Основные графические режимы для карты VGA приведены в табл. 1.

ТАБЛИЦА 1

Номер режима	Разрешение экрана	Количество цветов
12h	640 × 480	16
13h	320 × 200	256

Программирование видеокарты осуществляется через специальные 8-разрядные регистры. Доступ к регистрам осуществляется через порты ввода/вывода процессора. Регистры объединены в группы для соответствующего логического блока видеоадаптера. Каждой группе соответствует пара последовательных портов (порт адреса и порт значения). Для записи значения в регистр видеокарты необходимо сначала записать номер регистра в порт адреса, а затем записать значение в порт значения. Чтение регистров осуществляется по аналогичной схеме.

Основные (для программирования) логические блоки видеоадаптера и их регистры следующие.

СИНХРОНИЗАТОР (СЕКВЕНСЕР)

Синхронизатор генерирует внутренние синхросигналы для адресации видеопамяти. Регистры синхронизатора приведены в табл. 2.

ТАБЛИЦА 2

Адрес порта	Индекс регистра	Назначение регистра
3C4h		Регистры адреса
3C5h		Регистры данных:
	00h	Регистр сброса
	01h	Регистр режима синхронизации
	02h	Регистр маски карты (разрешение записи)
	03h	Регистр выбора знакогенератора
	04h	Регистр режима памяти

ГРАФИЧЕСКИЙ КОНТРОЛЛЕР

Графический контроллер обеспечивает обмен данными как между видеобуфером и центральным процессором, так и между видеобуфером и контроллером атрибутов. Регистры графического контроллера приведены в табл. 3.

Таблица 3

Адрес порта	Индекс регистра	Назначение регистра
-------------	-----------------	---------------------

Адрес порта	Индекс регистра	Назначение регистра
3CEh		Регистры адреса
3CFh		Регистры данных:
	00h	Регистр установки / сброса
	01h	Регистр разрешения установки/сброса
	02h	Регистр сравнения цвета
	03h	Регистр циклического сдвига данных/ выбора функции
	04h	Регистр выбора карты чтения
	05h	Регистр режима работы
	06h	Регистр смешанного вывода
	07h	Регистр безразличия цвета
	08h	Регистр битовой маски

КОНТРОЛЛЕР АТТРИБУТОВ

В адаптерах VGA контроллер атрибутов поддерживает 16-цветовую палитру. Адресный регистр контроллера атрибутов и 21 регистр данных подключены к порту 3C0h. В зависимости от состояния внутреннего триггера значение, записываемое в порт 3C0h, загружается либо в регистр адреса, либо в регистр данных. Для установки триггера следует произвести чтение регистра состояния видеоконтроллера (порт 3DAh – для цветного режима):

```
mov DX,3DAh
in AL,DX
```

Регистры контроллера атрибутов приведены в табл. 4.

ТАБЛИЦА 4

Адрес порта	Индекс регистра	Назначение регистра
3C0h		Регистр адреса атрибута
3C0h (для зап.) 3C1h (для чтен.)		Регистры данных:
	00h-0Fh	Регистры палитры
	10h	Регистр управления режимом
	11h	Регистр установки цвета рамки
	12h	Регистр разрешения цветовой плоскости
	13h	Регистр горизонтального сдвига изображения
	14h	Регистр выбора цвета

ВИДЕО-ЦАП

Видео-ЦАП имеет 256 18-разрядных регистра, в которых записаны 6-разрядные числа для RGB составляющих (таблица цветов). Видео-ЦАП преобразует 6-разрядное число для каждой составляющей в аналоговый сигнал. Регистры видео-ЦАП приведены в табл. 5.

ТАБЛИЦА 5

Адрес порта	Назначение регистра
3C6	Регистр маскирования пикселей
3C7	Регистр состояния ЦАП (для чтения)

3C7	Индекс читаемого регистра таблицы цветов (для записи)
3C8	Индекс записываемого регистра таблицы цветов
3C9	Регистр данных таблицы цветов

ШЕСТНАДЦАТИЦВЕТНЫЙ РЕЖИМ VGA

Для 16-цветового режима (12h) под каждый пиксель (точку) изображения отводится 4 бита видеопамати ($2^4 = 16$). Вся видеопамать карты делится на четыре равные части, называемые битовыми плоскостями. Причем, каждый бит пиксели расположен в своей битовой плоскости видеопамати (рис. 1), 1 байт которой соответствует 8 пикселям на экране.

Все битовые плоскости проецируются на один и тот же участок адресного пространства процессора, начиная с адреса A000 : 0.

Для работы с пикселем с координатами (x, y) необходимо определить адрес байта относительно начала страницы видеопамати, содержащий данный пиксель, и позицию пикселя внутри байта. Поскольку видеопамать под пиксели отводится последовательно слева направо и сверху вниз, то одна строка соответствует 80 байтам. В этом случае:

$$\text{адрес байта} = 80 * y + (x \text{ div } 8);$$

$$\text{номер бита} = 7 - (x \text{ mod } 8).$$

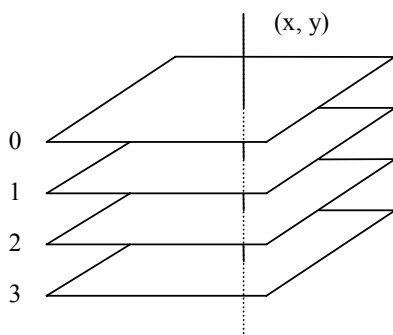


Рис. 1 Битовые плоскости видеокарты в 12h режиме

Битовые плоскости графического режима адаптера VGA адресуются параллельно. Когда процессор выполняет операцию чтения по адресу видеобуфера, из него пересылается не один, а четыре байта данных. Данные пересылаются не прямо в процессор, а в четыре 8-битовых регистра-защелки. Каждый подобный регистр соответствует своей битовой плоскости.

Аналогично выполняется операция записи точки в видеобуфер. В этом случае все четыре битовые плоскости могут изменяться параллельно с использованием комбинации данных, состоящей из содержимого регистров-защелок, байта данных пересылаемого процессором и предопределенного значения пикселя, хранящегося в регистре графического контроллера.

Возможны два режима чтения атрибутов точки видеобуфера (0 и 1) и четыре режима записи точки в видеобуфер (0, 1, 2 и 3). Выбор режима чтения определяется битом 3, а режим записи битами 0-2 регистра режима работы графического контроллера (3CF/05h).

РЕЖИМ ВИДЕОКАРТЫ VGA В РЕЖИМЕ 256 ЦВЕТОВ

В режиме 13h видеокарт одновременно отображает на экране 256 цветов при разрешении раstra 320*200. Для этого под каждый пиксель отводится 8 бит видеопамати. В данном режиме эти 8 бит идут последовательно один за другим, образуя один байт. В этом режиме плоскости видеопамати не используются. Видеопамать начинается с адреса A000 : 0. К точке с координатами (x, y) соответствует байт видеобуфера по адресу 320*y+x.

VESA BIOS EXTENTION ВИДЕОКАРТ SUPER VGA

После появления на рынке видеокарт с объемом памяти более предусмотренных стандартом 256 килобайт, появилась возможность создавать на экране изображения с лучшими, чем когда-либо до этого параметрами. Такие видеокарты стали называть видеокартами Super VGA.

Из-за того, что различные Super VGA продукты имеют разную аппаратную реализацию, прикладные программы испытывают значительные трудности при адаптации к каждой конкретной среде. Для исправления подобного неудобства в 1991 г. ассоциацией VESA был предложен одноименный стандарт на адаптеры Super VGA.

VESA BIOS Extention унифицирует подходы разработчиков графических приложений к программированию видеокарт Super VGA с различными аппаратными особенностями. Стандарт предлагает несколько функций для интерфейса с многообразием аппаратных реализаций Super VGA. Наиболее важная из них – функция 02: Установка видеорежима Super VGA. Эта функция освобождает прикладные программы от решения скучной и запутанной задачи установки видеорежима. Функция 05h обеспечивает интерфейс с аппаратной адресацией памяти. Функция 04h позволяет прикладным программам сохранять и восстанавливать состояние Super VGA без знания особенностей конкретной реализации.

ЛАБОРАТОРНАЯ РАБОТА 1

Программирование видеокарты VGA в режиме чтения 0

Цель работы: приобретение навыков в программировании видеокарты VGA в режимах чтения 0 для 16-цветового режима (12h).

Задание: организовать работу видеокарты VGA в режиме чтения 0. По заданным координатам изображения определить индекс цвета пикселя. Предложить наглядный интерфейс пользователя.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Для перехода видеокарты в режим чтения 0 необходимо 3 бит регистра режима работы графического контроллера (3CF/05h) установить в 0. Данный режим позволяет считывать байты из каждой отдельной битовой плоскости видеопамати. Значение, содержащееся в регистре выбора читаемой карты (3CF/04h), указывает на то, из какого регистра защелки будет производиться чтение байта данных (рис. 2). Для чтения пикселя с координатами x и y необходимо в регистр битовой маски (3CF/08h) записать номер интересующего бита, используя, например, следующее выражение $80h \text{ shr } (x \bmod 8)$. За четыре цикла чтения памяти видеобуфера по конкретному адресу ($80*y + (x \text{ div } 8)$) с изменяющимся номером читаемой карты (0, 1, 2 и 3) процессор получит значения всех регистров-защелок. И в каждом случае необходимо выделить соответствующий бит значения байта данных процессора и выделить соответствующий бит пикселя.

Содержание отчета

1. Дать описание данного режима.
2. Отобразить текст процедуры для чтения значения пикселя.
3. Оценить производительность представленной процедуры по отношению к стандартной GetPixel.

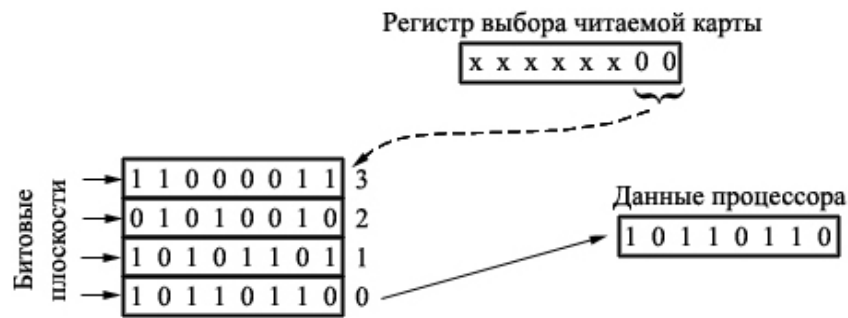


Рис. 2 Режим чтения 0

Лабораторная работа 2

ПРОГРАММИРОВАНИЕ ВИДЕОКАРТЫ VGA В РЕЖИМЕ ЧТЕНИЯ 1

Цель работы: приобретение навыков в программировании видеокарты VGA в режимах чтения 1 для 16-цветового режима (12h).

Задание: организовать работу видеокарты VGA в режиме чтения 1. Произвести поиск заданного значения пикселя на подготовленном растровом изображении. Предложить наглядный интерфейс пользователя.

Методические указания

В режиме чтения 1 каждое из восьми значений, записанных в регистры-защелки во время выполнения процессором операции чтения, сравнивается со значением, содержащимся в регистре сравнения цвета (02H). Результат этого сравнения возвращается в процессор как один байт (рис. 3). В случае совпадения одинаковых битов у значения пикселя и содержимого регистра сравнения цвета в передаваемом процессору байте данных устанавливается в 1 соответствующий бит. В случае несовпадения одинаковых битов соответствующий бит в байте данных содержит 0.

Следует обратить внимание на то, что сброс соответствующего бита в регистре безразличия цвета (3CF/07h) в 0 фактически исключает регистр-защелку из сравнения.

Например, значение 7 (00000111) в регистре безразличия цвета повлечет участие в операции сравнения только трех младших битов каждого значения пикселя.

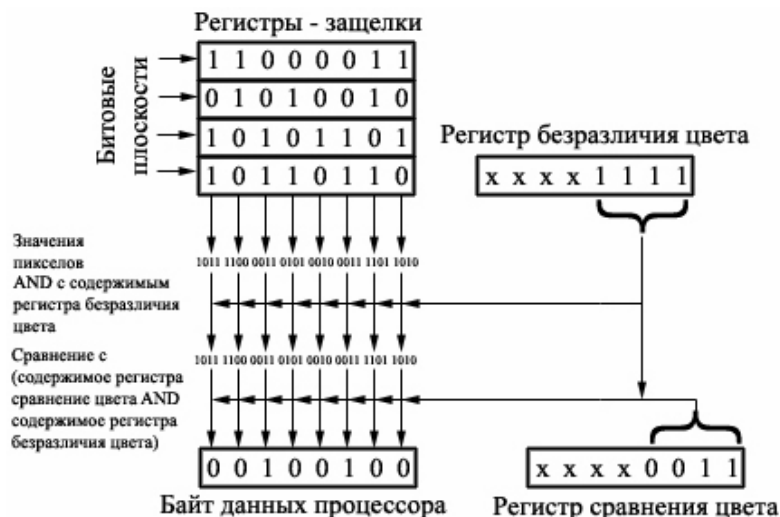


Рис. 3 Режим чтения 1
Содержание отчета

1. Дать описание данного режима.

2 Отобразить текст процедуры для поиска заданного значения пикселя в тестовом растровом изображении.

3 Предложить несколько ситуаций, где может быть использован данный режим чтения пикселей.

Лабораторная работа 3

Программирование видеокарты VGA в режиме записи 0

Цель работы: приобретение навыков в программировании видеокарты VGA в режимах записи 0 для 16-цветового режима (12h).

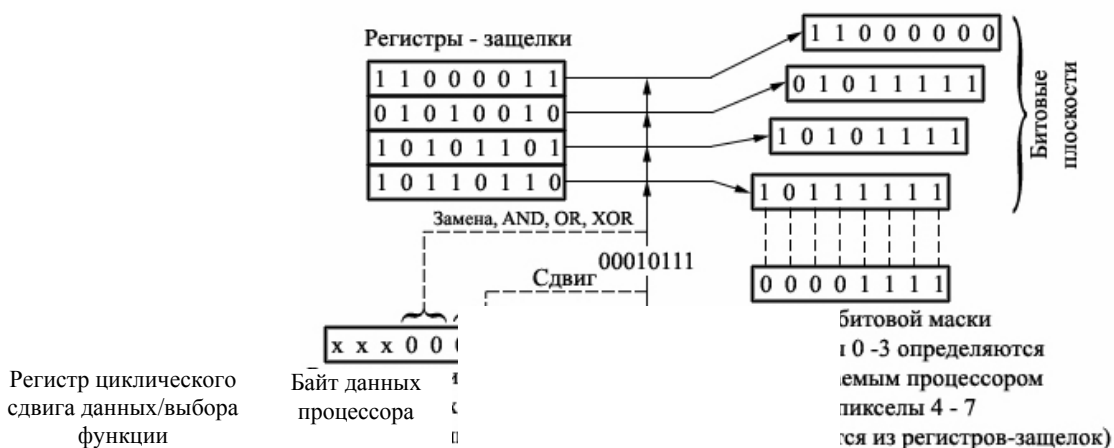
Задание: Организовать работу видеокарты VGA в режиме записи 0(a). Произвести вывод заданного значения пикселя в заранее подготовленное растровое изображение. Предложить наглядный интерфейс пользователя.

Методические указания

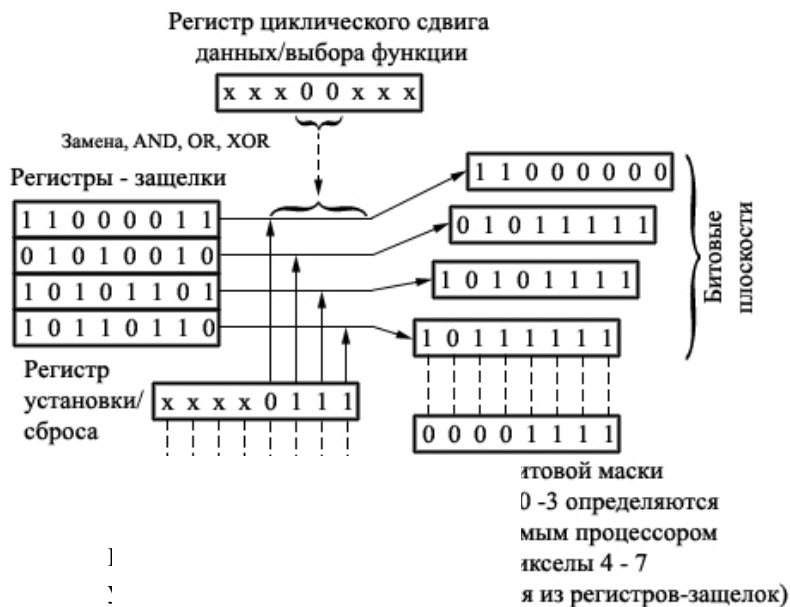
В режиме записи 0 производится комбинация байтовых и пиксельных операций, инициируемых в операции записи процессора. Байт данных, передаваемый процессором, может использоваться для изменения любой или всех битовых плоскостей; в то же время для модификации любого или всех восьми пикселей (хранящихся в регистрах-защелках) может привлекаться predetermined значение пикселя. Такая двухмерная модификация содержимого регистров-защелок осуществляется несколькими различными способами, определяемыми содержимым регистров установки/сброса (3CF/00h), циклического сдвига данных/выбора функции (3CF/03h) и регистра битовой маски (3CF/08h) (рис. 4).

Регистр битовой маски (3CF/08h) определяет, как образуется новое значение каждого из восьми пикселей в видеобуфере. Если бит в этом регистре равен 0, то соответствующее значение пикселя копируется в видеобуфер прямо из регистров-защелок. Для каждого единичного бита регистра битовой маски соответствующий пиксель получается комбинацией содержимого регистра-защелки с данными, поступающими из процессора или со значениями пикселя в регистре разрешения установки/сброса. Таким образом, если операции записи и чтения по одному и тому же адресу следуют друг за другом сразу же, то модифицируется только те пиксели, для которых соответствующий бит в регистре маски установлен в 1.

Регистр циклического сдвига данных/выбора функции (3CF/03h) содержит два поля, влияющих на способ модификации содержимого регистров-защелок. Биты 3 – 4 определяют, какая конкретно поразрядная логическая операция (AND, OR, XOR или замена) используется для получения значения пикселей (табл. 6). Биты 0 – 2 определяют число битов, на которое байт данных от процессора циклически сдвигается вправо перед объединением его с содержимым регистров-защелок.



a)



б)

Рис. 4 Режим записи 0. Значение регистра разрешения установки/сброса:

а – 1111В; б – 0000В

Таблица 6

Бит 4	Бит 3	Функция
0	0	Замена
0	1	AND
1	0	OR
1	1	XOR

Способ изменения содержимого битовых плоскостей (побайтно или попиксельно) определяется содержимым регистра разрешения установки/сброса (3CF/01h). Когда в этом регистре записано значение 0FH (1111B), каждый пиксель модифицируется соединением значения пикселя, содержащегося в регистре-защелке, с содержимым регистра установки/сброса (3CF/00h), причем логическая операция, используемая при объединении содержимого регистров, задается регистром сдвига данных/выбора функции (см. рис. 4, а). Когда регистр разрешения установки/сброса содержит 0, то сдвинутый байт данных процессора объединяется с байтами в каждой из защелок, а используемая логическая функция опять-таки определяется содержимым регистра сдвига данных/выбора функции (см. рис. 4, б). В любом случае модифицируются только пиксели, указываемые регистром битовой маски.

Примечание. В регистр разрешения установки/сброса можно записать любое значение – от 0 до 0FH. В этом случае каждый бит каждого пикселя модифицируется объединением его с соответствующим битом в регистре установки/сброса либо с соответствующим битом в байте данных процессора. Применяемый способ зависит от состояния соответствующего бита в регистре разрешения установки/сброса.

Содержание отчета

1. Дать описание данного режима.

- 2 Отобразить текст процедуры для записи заданного значения пикселя в тестовом растровом изображении.
- 3 Оценить производительность представленной процедуры по отношению к стандартной PutPixel.

Лабораторная работа 4

Программирование видеокарты VGA в режиме записи 1

Цель работы: приобретение навыков в программировании видеокарты VGA в режимах записи 1 для 16-цветового режима (12h).

Задание: организовать работу видеокарты VGA в режиме записи 1. Произвести копирования фрагмента растрового изображения. Предложить наглядный интерфейс пользователя.



Рис. 5 Режим записи 1

Методические указания

В режиме записи 1 при выполнении процессором операции записи содержимое регистров-защелок копируется прямо в битовые плоскости (рис. 5). На этот процесс не влияют ни значение байта данных процессора, ни содержимое регистров сдвига данных/выбора функции, битовой маски, установки/сброса и разрешения установки/сброса. Для выполнения операции в режиме записи 1 необходимо вначале осуществить операцию чтения процессора для инициализации регистров-защелок. Таким образом, за одну операцию перезаписи копируется сразу 4 байта (8 пикселей). Этот режим позволяет осуществлять быстрое копирование фрагментов видеопамати.

Содержание отчета

- 1 Дать описание данного режима.
- 2 Отобразить текст процедуры для копирования заданного фрагмента изображения в тестовом растровом изображении.
- 3 Оценить производительность представленной процедуры по отношению к аналогичным действиям стандартных растровых операций PutPixel и GetPixel.

Лабораторная работа 5

Программирование видеокарты VGA в режиме записи 2

Цель работы: приобретение навыков в программировании видеокарты VGA в режимах записи 2 для 16-цветового режима (12h).

Задание: организовать работу видеокарты VGA в режиме записи 2. Произвести запись значения точки в растровое изображение. Предложить наглядный интерфейс пользователя.

Методические указания

В режиме записи 2 младшие биты байта, пересылаемого процессором, играют ту же роль, что и содержимое регистра установки/сброса в режиме записи 0. Это значит, что содержимое битовых плоскостей модифицируется комбинированием значений пикселей в регистрах-защелках с данными процессора, причем используемая логическая операция задается содержимым регистра сдвига данных/выбора

функции (рис. 6). Как и в режиме записи 0, содержимое регистра битовой маски определяет, какие пиксели модифицируются с привлечением объединенных значений пикселей, а какие модифицируются непосредственно из регистров-защелок.

Содержание отчета

1. Дать описание данного режима.
2. Отобразить текст процедуры для записи значения пиксель по заданным координатам.
3. Оценить производительность представленной процедуры по отношению к аналогичным действиям стандартной операций PutPixel.

Лабораторная работа 6

Программирование видеокарты VGA в режиме записи 3

Цель работы: приобретение навыков в программировании видеокарты VGA в режимах записи 3 для 16-цветового режима (12h).

Задание: организовать работу видеокарты VGA в режиме записи 3. Произвести запись точки в растровое изображение. Предложить наглядный интерфейс пользователя.



Рис. 6 Режим записи 2
Методические указания

Режим записи 3 получается комбинированием содержимого регистров-защелок и содержимого регистра установки/сброса (3CF/00h). Байт данных процессора

циклически сдвигается на число битов, определяемое регистром сдвига данных/выбора функции (3CF/03h), и объединяется по AND со значением в регистре битовой маски (3CF/08h). Полученная битовая маска здесь играет роль, аналогичную значению регистра битовой маски в режимах записи 0 и 2, т.е. она определяет, какие пиксели в битовых плоскостях модифицируются объединением значений пикселей из регистров-защелок с содержимым регистра установки/сброса (3CF/00h), а какие модифицируются непосредственно из регистров-защелок (рис. 7).

Содержание отчета

1. Дать описание данного режима.
2. Отобразить текст процедуры для записи значения пиксель по заданным координатам.
3. Оценить производительность представленной процедуры по отношению к аналогичным действиям стандартной операций PutPixel.

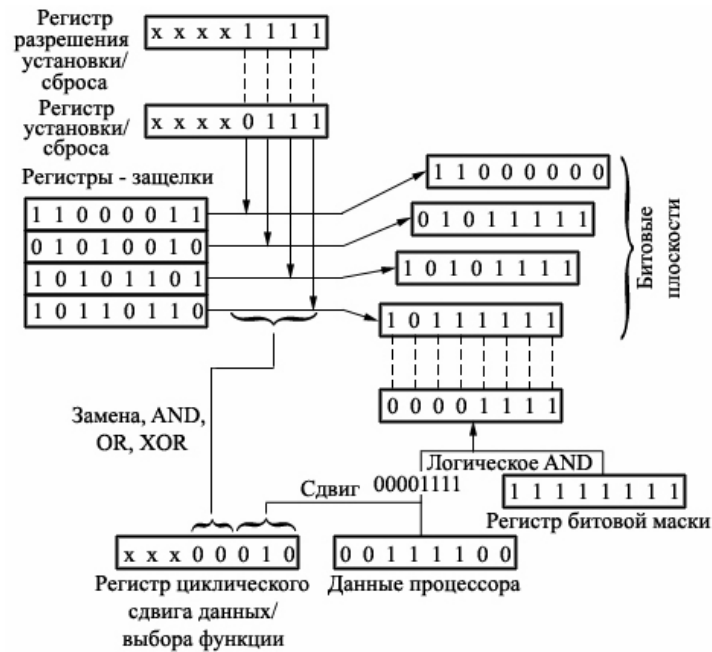


Рис. 7 Режим записи 3

Лабораторная работа 7

Программирование видеокарты VGA в режиме 13h

Цель работы: приобретение навыков в программировании видеокарты VGA в режимах режиме 13h (320 × 200 256 цветов).

Задание: организовать работу видеокарты VGA в режиме 13h. Произвести вывод и запись пикселя в растровое изображение в данном режиме. Предложить наглядный интерфейс пользователя.

Методические указания

Для перехода в 13h h можно воспользоваться следующим кодом программы.

```

procedure InitMode;
var
  R: Registers;
begin
  R.ah := 0; {установка режима}
  R.al := $13; {номер режима}
  Intr($10, R);
end;

```

Режим 13h обеспечивает 320 × 200 точек в растровом изображении с использованием одновременно 256 цветов. Изображение размещается в одной байтовой плоскости видеобuffers. Видеопамять начинается с адреса A000 : 0. Пиксель с координатами (x, y) соответствует байту видеопамяти по адресу A000 : (320*y + x). И, например, для записи значения пикселя по координате x и y в видеобuffer достаточно выполнить процедуру

```

procedure Pixel(x, y: Integer; Color: Byte);
begin

```

```
Mem[$A000:320*y+x] := Color  
end;
```

Переменная Color может принимать значения 0 – 255.

Содержание отчета

1. Дать описание данного режима.
2. Отобразить текст процедуры для записи значения пикселя в режиме 13h по заданным координатам.
3. Дать свое заключение о качестве построенных изображений.

ЛАБОРАТОРНАЯ РАБОТА 8

Программирование видеокарты SuperVGA в стандарте VESA 1.2

Цель работы: приобретение навыков в программировании видеокарты Super VGA в стандарте VESA 1.2.

Задание: организовать работу видеокарты Super VGA в стандарте VESA 1.2.

Для тестируемой видеокарты показать список поддерживаемых режимов VESA. Выбрать один из графических режимов и в нем продемонстрировать вывод растровой графики. Предложить наглядный интерфейс пользователя.

Методические указания

Номера стандартных VGA-режимов имеют ширину 7 битов и находятся в диапазоне от 00h до 13h. На конкретных видеокартах определены номера видеорежимов от 14h до 7Fh. Значения в диапазоне от 80h до FFh не могут быть использованы, поскольку функция 00h VGA BIOS (установка видеорежима) интерпретирует бит 7 как флаг очистки видеопамати.

Из-за ограничения в 7 бит для номеров режимов, видеорежимы VESA имеют ширину в 15 битов. Для инициализации Super VGA режима его номер помещается на регистр BX при вызове функции 02h VESA BIOS (Установка Super VGA режима).

Номера режимов VESA имеют следующий формат:

D0-D8 = Номер режима

Если D8 == 0, это режим, определенный VESA

Если D8 == 1, это не-VESA режим

D9-D14 = Зарезервировано VESA для будущих расширений (= 0)

D15 = Зарезервирован (= 0)

Таким образом, номера VESA-режимов начинаются со 100h. При такой схеме нумерации номера стандартных VGA-режимов и режимов, определенных для конкретных видеоплат, реализуются как подмножество номеров VESA. Это означает, что обычные режимы VGA могут инициализироваться с помощью функции 02h VESA (установка Super VGA режима), за счет простого помещения номера режима на BL и очисткой старшего байта.

В настоящее время VESA определила 7-битный номер режима 6Ah для графического 4-слойного 16-цветного режима с разрешением 800 × 600. Соответствующий 15-битный номер режима для этого режима 102h.

Определены следующие номера VESA режимов для графики (табл. 7).

Работа с VESA BIOS Extensions (VBE) заключается в вызове подфункций функции 4Fh прерывания 10h. Состав основных подфункций стандарта VESA 1.2 представлен в табл. 8.

ТАБЛИЦА 7

Но- мер	Разре- шение	Цвета	Но- мер	Разре- шение	Цвета
------------	-----------------	-------	------------	-----------------	-------

ре- жима			ре- жима		
100h	640 × 400	256	111h	640 × 480	64K (5:6:5)
101h	640 × 480	256	112h	640 × 480	16.8M (8:8:8)
102h	800 × 600	16	113h	800 × 600	32K (1:5:5:5:)
103h	800 × 600	256	114h	800 × 600	64K (5:6:5)
104h	1024 × 768	16	115h	800 × 600	16.8M (8:8:8)
105h	1024 × 768	256	116h	1024 × 768	32K (1:5:5:5:)
106h	1280 × 1024	16	117h	1024 × 768	64K (5:6:5)
107h	1280 × 1024	256	118h	1024 × 768	16.8M (8:8:8)
10Dh	320 × 200	32K (1:5:5:5:)	119h	1280 × 1024	32K (1:5:5:5:)
10Eh	320 × 200	64K (5:6:5)	11Ah	1280 × 1024	64K (5:6:5)
10Fh	320 × 200	16.8M (8:8:8)	11Bh	1280 × 1024	16.8M (8:8:8)
110h	640 × 480	32K (1:5:5:5:)			

Таблица 8

Значение регистра ах	Название подфункции	Описание параметров
ах = 4F00h	GetVESAInfo Получить информацию расширенного BIOS	es: di – адрес буфера для информации выходной параметр: al = 4F – данная функция поддерживается видеоадаптером ah = 00 – успешное завершение ah = 01 – ошибка
ах = 4F01h	GetVESAModeInfo Получить информацию о режиме	сх – VBE режим, es: di – адрес буфера для информации
ах = 4F02h	SetVESAMode Установить режим	bx – видеорежим. Можно использовать стандартные VGA режимы
ах = 4F03h	GetVESAMode Получить текущий видеорежим	выходной параметр: bx – видеорежим
ах = 4F04h	Set/GetVideoState Устано-	Получить размер буфера dl = 0 сх = 0Fh

Значение регистра ax	Название подфункции	Описание параметров
	вить/сохранить информацию VBE	выходной параметр: bx – размер буфера для сохранения информации VBE Сохранить информацию dl = 1 cx = 0Fh es: bx – буфер Установить информацию dl = 1 cx = 0Fh es: bx – буфер

Продолжение табл. 8

Значение регистра ax	Название подфункции	Описание параметров
ax = 4F05h	Set/GetVideoBank Установка видеобанка	bx = 0 dx – номер банка bx = 100h выходной параметр: dx – номер банка
ax = 4F06h	Установить/прочитать логическую длину скан-строк	См. описание функции
ax = 4F07h	Установить/прочитать начало высвечивания	См. описание функции
ax = 4F08h	Установить/прочитать информацию о DAC-палитре	См. описание функции

Функция 4F00h GetVESAInfo

Эта функция предназначена для получения информации о возможностях видеоадаптера. Для этого используется следующая структура данных (табл. 9).

Таблица 9

Смещение	Размер	Описание
00h	4 байта	Сигнатура "VESA"
04h	WORD	Версия VESA
06h	DWORD	Указатель на ASCIIZ строку производителя
0Ah	4 байта	Возможности: Бит 0 – установлен, если DAC может

Смещение	Размер	Описание
		изменять ширину, и очищен, если он содержит 6 бит на основной цвет.
0Eh	DWORD	Указатель на список поддерживаемых режимов. Список представляет массив двухбайтовых идентификаторов, заканчивающийся значением 0FFFFh.
12h	WORD	Размер видеопамяти в блоках по 64 килобайта
14h-100h		Резерв

Эта структура может быть представлена в виде записи (record) языка Pascal следующим образом:

Type

TVESAInfo = record

Signature : array [0..3] of char;

Version : word;

OEM : Pchar;

Capabilities: longint;

Modes : ^Word;

VideoMem : Word;

Reserved : array [\$14..\$FF] of byte;

End;

Ф У Н К Ц И Я 4 F 0 1 H G E T V E S A M O D E I N F O

Эта функция предназначена для получения подробной информации о конкретном видеорежиме. Структура данных VESAModeInfo представлена в табл. 10.

Таблица 10

Смещение	Размер	Описание
00h	WORD	Атрибуты режима: Бит 0 – если установлен, то режим поддерживается Бит 1 – если установлен, то доступна дополнительная информация Бит 2 – поддерживается режим вывода через BIOS Бит 3 – если установлен, то цветной, иначе монохромный Бит 4 – если установлен, то графический, иначе текстовый Биты 5 – 7 резерв
02h	Байт	Атрибуты окна A: Бит 0 – установлен, если окно существует Бит 1 – окно можно читать Бит 2 – в окно можно выводить Биты 3 – 7 резерв

03h	Байт	Атрибуты окна В
04h	WORD	Размер сегмента окна
06h	WORD	Размер окна
08h	WORD	Начальный сегмент для окна А
0Ah	WORD	Начальный сегмент для окна В
0Ch	DWORD	Дальний указатель на функцию позиционирования окна (см. 4F05h)
10h	WORD	Байт на скан-строку

Продолжение табл. 10

Смещение	Размер	Описание
<i>СЛЕДУЮЩАЯ ИНФОРМАЦИЯ НЕОБХОДИМА ТОЛЬКО ДЛЯ OEM-РЕЖИМОВ</i>		
12h	WORD	Ширина в пикселях
14h	WORD	Высота в пикселях
16h	Байт	Ширина знакоместа в пикселях
17h	Байт	Высота знакоместа в пикселях
18h	Байт	Количество плоскостей видеопамяти
19h	Байт	Количество бит на пиксель
1Ah	Байт	Количество банков
1Bh	Байт	Тип модели памяти: 0 – текст 1 – графика CGA 2 – графика Геркулес 3 – графика EGA 16 цветов 4 – пакованные пиксели 5 – байтовые 256 цветные режимы (без чередования) 6 – прямой режим 15/16/24 бита на цвет 7 – цветояркий режим YUV 8-0Fh – зарезервированы под VESA 10h-0FFh – зарезервированы под режимы OEM
1Ch	Байт	Размер банка в килобайтах
1Dh	Байт	Количество страниц изображения
1Eh	Байт	Зарезервировано

Пример реализации данной функции:

```
function GetVESAModeInfo (Mode: word; Buffer: pointer): boolean;
var RetCode: word;
begin
  asm
    mov ax, $4f01
    mov cx, Mode
    les di, Buffer
    int $10
    mov RetCode, ax
  end;
  GetModeInfo := RetCode = $004F;
```

end;

Ф У Н К Ц И Я 4 F 0 2 H S E T V E S A M O D E

Функция устанавливает требуемый видеорежим.

```
function SetVESAMode(Mode: word): boolean;  
var RetCode: word;  
begin  
  asm  
    mov ax, $4f02  
    mov bx, mode  
    int $10  
    mov RetCode, ax  
  end;  
  SetVESAMode := RetCode = $004f;  
end;
```

Ф У Н К Ц И Я 4 F 0 3 H G E T V E S A M O D E

Эта функция прямо противоположна предыдущей – она возвращает текущий видеорежим.

Ф У Н К Ц И Я 4 F 0 4 H S E T / G E T V I D E O S T A T E

Функция предназначена для манипулирования с режимами отображения. Она полезна, если приложению необходимо часто переключаться в процессе работы между видеорежимами.

Вход: AH = 4Fh Поддержка Super VGA
 AL = 04h Сохр./восст. состояния Super VGA
 DL = 00h Возврат размера буфера сохр./восст. состояния
 CX = Требуемое состояние
 D0 = Сохр./восст. аппаратного состояния
 D1 = Сохр./восст. состояния данных видео BIOS
 D2 = Сохр./восст. состояния видео-DAC
 D3 = Сохр./восст. состояния Super VGA

Выход: AX = Состояние
 BX = Число 64-байтовых блоков в буфере сохранения
 Все другие регистра сохраняются

Вход: AH = 4Fh Поддержка Super VGA
 AL = 04h Сохр./восст. состояния Super VGA
 DL = 01h Сохранение видео состояния Super VGA
 CX = Требуемое состояние (см. выше)
 ES: BX = Указатель на буфер

Выход: AX = Состояние
 Все другие регистры сохраняются

Вход: AH = 4Fh Поддержка Super VGA
 AL = 04h Сохр./восст. состояния Super VGA
 DL = 02h Восстановление видео состояния Super VGA
 CX = Требуемое состояние (см. выше)
 ES: BX = Указатель на буфер

Выход: AX = Состояние
 Все другие регистры сохраняются

Ф У Н К Ц И Я 4 F 0 5 H S E T / G E T V I D E O B A N K

Функция выполняет позиционирование по видеопамяти. Она необходима, если общее количество точек в данном видеорежиме превышает 65536 (64 К).

```
function SetVideoBank (Bank, Window: word): boolean;
```

```

var RetCode: word;
begin
  if Bank = NumBank then
    SetVESABank := TRUE
  else begin
    asm
      mov ax, $4f05
      mov bx, Window
      mov dx, Bank
      int $10
      mov RetCode, ax
    end;
    if RetCode = $004f then begin
      SetVESABank := TRUE;
      NumBank := Bank;
    end
    else SetVESABank := FALSE;
  end;
end;

```

Функция 4F06h

Данная функция устанавливает длину логических строк раstra. Эта функция позволяет прикладной программе устанавливать размер видеобуфера более широким, чем высвечиваемая область. Функция 07h позволяет затем указывать в таком буфере начало высвечиваемой области.

Функция 4F07h

Эта функция выбирает пиксель логической страницы, который высвечивается в левом верхнем углу экрана. Эта функция может использоваться для скроллинга логического экрана, размеры которого превышают размер высвечиваемого экрана. Эта функция может также использоваться для быстрого переключения между двумя высвечиваемыми экранами для создания эффекта анимации с помощью двух буферов.

Функция 4F08h

Эта функция опрашивает и устанавливает режим функционирования DAC-палитры. Ширина палитры DAC сбрасывается в стандартное для VGA 6-битное значение при вызове стандартной или VESA (AX = 4F02h) функции установки режима.

СХЕМА ПРОГРАММИРОВАНИЯ ВИДЕОКАРТЫ SUPER VGA В РЕЖИМЕ VESA 1.2

1) Сначала программа должна разместить буфер длиной в 256 байтов. Этот буфер будет использоваться VESA BIOS для возврата информации о видеосреде.

2) Программа должна затем вызвать функцию 00h из VESA BIOS (Возврат информации о Super VGA). Если регистр AX не равен 004Fh при возврате из этой функции, то программа должна определить, что VESA BIOS Extension отсутствует и не поддерживать такую ситуацию.

Если регистр AX не содержит кода ошибки, вызов функции был успешен. VESA BIOS Extension заполняет буфер разнообразной информацией. Программа может проверить правильность VESA-блока, идентифицируя символы "VESA" в начале блока. Программа может также проверить поле VESAVersion для того, чтобы определить обладает ли VESA BIOS Extension достаточными функциональными возможностями. Программа может использовать OEMStringPtr для определения информации, специфичной для конкретной видеоплаты.

Наконец, программа может получить список поддерживаемых Super VGA режимов, используя VideoModePtr. Это поле содержит указатель на список видеорежимов, поддерживаемых видеосредой.

3) Программа должна затем создать новый буфер и вызвать из VESA BIOS функцию 01h (Возврат информации о режиме Super VGA), чтобы получить информацию о поддерживаемых видеорежимах.

Используя VideoModePtr, полученный на шаге 2, программа должна вызывать эту функцию до тех пор, пока не будет найден подходящий видеорежим. Если надлежащий видеорежим не найден, программа не поддерживает такую ситуацию.

Функция возврата информации о режиме Super VGA заполняет буфер, снабжая тем самым программу информацией, описывающей возможности видеорежима. Блок данных содержит всю необходимую информацию, позволяющую программе эффективно использовать возможности этого режима.

Программа должна проверить поле ModeAttributes. Чтобы проверить, что режим действительно поддерживается, программа должна проверить бит D0. Если он равен 0, то режим не поддерживается аппаратурой.

4) После выбора программой видеорежима следующий шаг состоит в его инициализации. Однако, программа может пожелать сначала сохранить текущий видеорежим. Когда программа закончится, этот режим будет восстановлен. Для получения текущего видеорежима должна использоваться функция 03h (Чтение режима Super VGA).

5) Для инициализации видеорежима программа должна использовать функцию 02h VESA BIOS (Установка режима Super VGA).

6) Перед окончанием программа должна восстановить первоначальный видеорежим. Этот режим, полученный на шаге 4, может быть либо стандартным VGA-режимом, либо специфичным для конкретной платы, либо поддерживаемым VESA-режимом. Он должен быть инициализирован с помощью функции 02h VESA BIOS (Установка режима Super VGA). На этом программа завершает свою работу.

Пример рисования точки в текущем видеорежиме (256 цветов)

```
procedure Pixel(X, Y: Word; Color: Byte);
var
  Nb: Word;
  Sm: Word;
  Adr: Longint;
begin
  {получим абсолютное смещение относительно начала видеопамати}
  {BytePerScanLine – длина строки раstra в байтах}
  Adr := X + Y * BytePerScanLine;
  {получим номер банка видеопамати}
  {WindowAGranularity – размер окна}
  Nb := Adr div WindowAGranularity;
  {получим смещение относительно начала окна}
  Sm := Adr mod WindowAGranularity;
  {установим требуемый банк}
  asm
    mov ax,4F05h
    mov bx,0
    mov dx,Nb
  end;
  {выведем точку}
  MemB[$A000:Sm] := Color;
end;
```

Содержание отчета

- 1 Дать описание алгоритма программирования видеокарт в режиме VESA 1.2.
- 2 Отобразить текст процедур для записи и чтения значения пиксель в стандарте VESA 1.2.
- 3 Дать свое заключение о качестве построенных изображений.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

- 1 Шикин Е.В., Боресков А.В. Компьютерная графика. Полигональные модели. М.: Диалог МИФИ, 2000. 461с.

2 Уилтон Р. Видеосистемы персональных компьютеров IBM PC и PS/2: Руководство по программированию. М.: Радио и связь, 1994. 383с.