

Тема 5 Video BIOS. Функції VGA BIOS

Як вже наголошувалося, всі операції по модифікації вмісту відеопам'яті і регістрів відеоадаптера VGA виконує ЦП. Реалізація будь-якої графічної операції (зміна відеорежимів, читання/запис інформації в кадровий буфер, управління курсором і т.д.) вимагає від ЦП виконання вельми довгої послідовності елементарних команд (зчитування/запис, переміщення даних в регістрах і ін.). З метою спрощення програмування відеоадаптера, а також для забезпечення сумісності апаратного і програмного забезпечення всі набори команд ЦП, які реалізують графічні функції, поміщаються в спеціальний ПЗП, розташований на платі відеоадаптера VGA. Ці підпрограми утворюють розширення базової **системи** введення-виведення і називаються Video BIOS.

Фактично **Video BIOS** – це набір підпрограм, написаних в кодах команд ЦП і призначених для реалізації основних функцій відеосистеми. Набір цих функцій назвали **відеосервіс**. Крім того, Video BIOS містить безліч даних, необхідних для роботи в різних відеорежимах, дані про виробника, модель і можливостях відеоадаптера. Video BIOS зберігається в спеціальному ПЗП (ROM), його місткість складає 32 Кбайт. ROM Video BIOS відеоадаптера VGA є 8- розрядним. Низька швидкодія ПЗП, в порівнянні з оперативною пам'яттю ПК, уповільнює роботу відеосистеми. Для прискорення роботи практично у всіх ПК передбачена можливість копіювання (за допомогою програми конфігурації CMOS Setup) вмісту ROM BIOS і ROM Video Bios в спеціально відведену область оперативної пам'яті ПК, названу тінювою пам'яттю. Використовування механізму тінювої **пам'яті збільшує** швидкодію системи, а також дозволяє при необхідності програмним шляхом замінити Video BIOS на більш нову версію.

Відеосервіс BIOS

Відеосервіс BIOS реалізується за допомогою механізму програмних переривань, тобто шляхом видачі прикладною програмою ЦП команди INT на переривання з вказівкою номера переривання. Архітектурою ПК передбачені апаратні і програмні переривання. Апаратні переривання ініціалізувалися різними пристроями ПК шляхом подачі на ЦП електричного сигналу по спеціальних дротах, названих лініями запиту на переривання IRQ.

Програмні переривання реалізуються шляхом подачі прикладною програмою команди INT центральному процесору. Перед подачею цієї команди прикладна програма заносить номер переривання в регістр AX (1-ий з регістрів загального призначення). Звичайно, в рамках одного переривання доступно декілька функцій, при цьому окремі функції можуть містити декілька підфункцій. Для вибору необхідної функції (підфункції) її номер перед викликом переривання також заноситься в відповідний регістр загального призначення ЦП. Номер переривання однозначно пов'язаний з адресою програми обробки переривання, яку повинен викликати ЦП при ініціалізації переривання. Для простої і ефективної локалізації процедури обслуговування переривань слугує таблиця векторів переривань, що міститься в перших елементах оперативної пам'яті.

Перш ніж починати роботу в якому-небудь режимі, необхідно спочатку перевести систему в цей режим. Проте, зробити це не просто. Категорично не рекомендується програмістам, які починають працювати, ставити експерименти по прямому управлінню режимами роботи відеоконтроллера через його регістри, тому що некоректна установка параметрів може вивести з ладу монітор. Для відеоконтроллерів розділення операцій на апаратні і програмні є абсолютно жорстким: будь-які перемикання режимів повинні виконуватися за допомогою функцій BIOS відеоконтроллера або фірмових драйверів, а виводити інформацію потрібно напряму у відеопам'ять. BIOS володіє великими можливостями: вона забезпечує реалізацію всіх можливостей відеосистеми. Очищення і прокрутка екрану, позиціонування курсору, зміна кольору символів, завантаження шрифту користувача, зміна відеосторінок і відеорежимів, навіть зміна форми курсору – все це доступно за допомогою функцій

переривання BIOS 10h. Для програм BIOS характерна більш висока швидкість роботи. Проте за все це доводиться платити помітним ускладненням програми.

Встановлювати необхідний відеорежим треба 1 раз при запуску прикладної програми – на весь період її виконання. Перемикаючи режими в процесі роботи у край небажано: при цьому спостерігаються різні відеоефекти (збій синхронізації, “сніг”, мерехтіння і т.д.). Тому відеорежим повинен дозволяти виконувати все те, що вимагається в процесі операції – якщо потрібно виведення текстової і графічної інформації, то необхідно відразу встановити графічний режим.

Існують два основні способи програмної установки відеорежиму: за допомогою функцій VGA BIOS і за допомогою функцій VESA BIOS.

Функції VGA BIOS

Графічні режими VGA, за час що пройшов з моменту створення (1987г.) цього типу відеоконтролерів зараз сильно застаріли, а текстові використовуються 20 років і продовжують успішно застосовуватися (завдяки простоті виконання операцій в цих режимах). Тому зараз інтерес для програмістів представляють тільки невелика підгрупа функцій із стандартного набору VGA BIOS, призначена для роботи в текстовому режимі:

- установка режиму;
- управління положенням і розміром курсору;
- перемикання сторінок;
- управління шрифтами;

Для виклику функцій VGA BIOS використовується переривання INT 10h. Набір функцій дуже великий, але в цілому застарілий. Розглянемо тільки ті функції, які доцільно застосовувати і зараз.

Більш детально в наступній літературі:

1. В Браун Р., Кайл Дж “Довідник по перериваннях для IBM PC”
2. В Уїлсон Р. “Відеосистеми персональних комп'ютерів IBM PCU, PS/2”
3. Фролов А.В., Фролов Г.В. “Програмування відеоадаптерів CGA, EGA і VGA”

BIOS надає користувачу ряд функцій для управління дисплеєм. Ці функції викликаються через переривання 10h. Всі вони перераховані в таблиці

Таблиця 2.1

Номер	Функція
0 (00h)	установка відеорежиму дисплея
1 (01h)	установка типу курсору
2 (02h)	встановити позицію курсору
3 (03h)	отримати позицію і розмір курсору
4 (04h)	не використовується
5 (05h)	встановити відео сторінку
6 (06h)	Прокрутка вікна вгору
7 (07h)	Прокрутка вікна вниз
8 (08h)	Прочитати символ і атрибут у позиції курсору
9 (09h)	вивести символ і атрибут в позицію курсору
10 (0Ah)	запис символу в позицію курсору
11 (0Bh)	установка монітора
12 (0Ch)	запис пікселя
13 (0Dh)	Читання пікселя
14 (0Eh)	вивести символ в режимі телетайпу
15 (0Fh)	отримання режиму
16 (10h)	установка регістрів палітри регістрами RAMDAC з атрибутом яскравості (мерехтіння)

17(11h)	установка знакогенератора
18(12h)	додаткові функції підтримки EGA
19(13h)	виведення рядка в режимі телетайпу
20(14h)	завантаження фонтів VGA
21(15h)	отримати фізичні параметри активного дисплея
26 (1Ah)	отримання інформації про стан відео системи
27 (1Bh)	збереження
28 (1Ch)	Стан відео системи

Функції з номерами 0-15 підтримуються будь-яким адаптером, що допускає графіку. ПЗП, що містить відеофункції переривання 10h знаходиться на платі відеоадаптера і, природно, коректно управляє відеосистемою. Це є однією з головних переваг використання 10-го переривання. Недоліком є повільна їх робота. Проте такі функції, як установка режиму або перемикавання відеосторінок, будуть корисні в будь-якій програмі.

За кожною функцією відеосервісу BIOS закріплені свої регістри загального призначення. Даний відеосервіс забезпечує просте і надійне управління відеосторінкою ПК, незалежно від її конкретної апаратної реалізації. У нього 1 істотний недолік – низька швидкість виконання функцій. З метою підвищення швидкодії відеосистеми використовують безпосередню зміну стану регістрів відеоадаптера прикладною програмою. Такий метод є більш швидким і ефективним, але вимагає детального знання апаратних особливостей відеоадаптера, що використовується (адрес портів введення-виведення, призначення і структури даних в управляючих регістрах і т.д.).

Управління відеоадаптерами шляхом модифікації вмісту його регістрів – це робота на свій страх і ризик, оскільки щонайменша помилка може привести до збоїв в роботі відеосистеми, до зависання комп'ютера, а в деяких випадках – до виходу з ладу монітора.

Переривання Int 10h, функція 00h: встановити відеорежим.

Дана функція призначена для встановлення заданого відеорежиму. Коди VGA – режимів приведені в наступній таблиці

Таблиця 2.2 Призначення рядкових операцій

Код режиму	Тип режиму	Дозвіл	Число кольорів	Адреса буфера
00h	Текс. колір*	40*25	16	B8000h
01h	Текс. Колір	40*25	16	B8000h
02h	Текс. колір*	80*25	16	B8000h
03h	Текс. Колір	80*25	16	B8000h
04h	Граф. колір	320*200	4	B8000h
05h	Граф. колір*	320*200	4	B8000h
06h	Граф. колір	640*200	2	B8000h
07h	Текст. монох.	80*25	2	B8000h
0Dh	Граф. колір.	320*200	16	A000h
0Eh	Граф. колір.	640*200	16	A000h
0Fh	Граф. монох.	640*350	2	A000h
10h	Граф. колір.	640*350	16	A000h
11h	Граф. монох.	640*480	2	A000h
12h	Граф. колір.	640*480	16	A000h
13h	Граф. колір.	320*200	256	A000h

*- В даному режимі відключений механізм перетворення палітри.

Всі сучасні відеоконтролери підтримують текстовий режим, а також зберігають сумісність із застарілими графічними режимами VGA на той випадок, якщо користувачу знадобиться запустити яку-небудь із старих програм. Перед викликом переривання вимагається занести в регістри наступні значення:

- в AH – значення 00h;
- в AL – код відеорежиму;

Функцію установки текстового режиму звичайно потрібно викликати при входженні в програму, яка працює в цьому режимі (оскільки невідомо в якому режимі працювала DOS перед запуском програми). Функцію також викликають перед поверненням в DOS з програми, що використала який-небудь графічний режим – щоб не було проблем із запуском подальших програм. Даний приклад ілюструє роботу з перериванням Int 10h – перехід в графічний режим і виведення на екран горизонтальної лінії.

Приклад 2.1 Вивести на екран горизонтальну лінію, користуючись перериванням;
; Встановимо графічний режим:

```
mov AH,00h      ; (1) Функція завдання режиму;
mov AL,10h      ; (2) Графічний режим;
int 10h         ; (3) Виклик BIOS;
```

; Намалюємо пряму лінію в циклі по координаті x:

```
mov SI,150      ; (4) Початкова X- координата;
mov CX,300      ; (5) Число точок по горизонталі;
line: push CX   ; (6) Збережемо його в стеку;
mov AH,0Ch      ; (7) Функція виведення пікселя;
mov AL,4        ; (8) Колір червоний;
mov BH,0        ; (9) Відеосторінка;
mov CX,SI       ; (10) X- координата (змінна);
mov DX,175      ; (11) y- координата (константа);
int 10h         ; (12) Виклик BIOS;
inc SI          ; (13) Інкремент x- координат
pop CX          ; (14) Відновимо лічильник кроків;
loop line       ; (15) Цикл з CX- кроків
```

; Зупинимо програму для спостереження результату її роботи:

```
mov AH,08h      ; (16) Функція введення з клавіатури без еха;
int 21h         ; (17) Виклик DOS;
```

; Перемкнемо відеоадаптер назад в текстовий режим:

```
mov AH,00h      ; (18) Функція завдання режиму;
mov AL,03h      ; (19) Текстовий режим;
int 10h         ; (20) Виклик BIOS;
```

В пропозиціях 1-3 за допомогою функції 00h переривання BIOS 10h здійснюється переведення відеоадаптера в графічний режим. Оскільки, номер режиму заноситься в байтовий регістр AL, всього можна здійснити 256

різних текстових і графічних режимів, з яких на сьогоднішній день використовуються близько 100. Режим 10h забезпечує виведення графічного зображення 16 кольорами з дозволом 640*350 точок і широко використовується при роботі з відео адаптерами EGA і VGA.

Зображення малюється по точках (в BIOS не передбачено програмних засобів виведення яких-небудь геометричних фігур або хоча б ліній, як немає і засобів заповнення областей екрану). Для висновку на екран кольорової точки використовується функція 0Ch переривання 10h. Ця функція вимагає занесення в регістр AL коду кольору, а в BH- номери відеосторінки, в CX- X-координати точки, що виводиться, в діапазоні 0..349, а в DX- y-

координати точки в діапазоні 0..639. Оскільки регістр CX вже використовується як лічильник кроків в циклі, для зберігання x-координати зарезервований регістр SI.

Пряма лінія в даному прикладі малюється шляхом виклику функції OCh в циклі, на кожному кроці якого значення y-координати залишається незмінним (175), а значення x-координати збільшується на 1 (13-а пропозиція). Після завершення циклу формування зображення в програмі передбачена зупинка (речення 16 и 17) для того, щоб користувач міг, залишаючись в графічному режимі, проаналізувати результати роботи програми. Для зупинки програми використовується функція DOS 08h введення одного символу з клавіатури. Вона не відображає введений символ на екран і тим самим не спотворює графічне зображення. Натиснення будь-якої клавіші (окрім управляючих Ctrl, Alt, Shift.) відновлює виконання програми.

В кінці розглянутого фрагмента передбачений переведення відеоадаптера в стандартний текстовий режим з номером 03h (пропозиція 18-20). Якщо такий перехід не виконати, відеоадаптер залишиться в графічному режимі, це може заважати правильному виконанню прикладних програм.

Переривання Int 10h, функція 01h: встановити розмір курсору.

Функція призначена для задання розміру курсору в текстовому відеорежимі. Застосовується, вона, звичайно, в текстових редакторах для повідомлення користувача про поточний режим (вставка або заміщення символів). Перед викликом переривання потрібно занести в регістри наступні значення:

в AH – код 01h

в CH – номер початкового рядка курсору в знакомісці

в CL – номер кінцевого рядка курсору в знакомісці

Номер початкового і кінцевого рядка курсору задаються щодо верхньої межі знакомісця. Для задання номерів початкового і кінцевого рядків в регістрах CH і CL використовуються тільки розряди 0-4, в розрядах 5-7 потрібно записати нулі. Примітка: курсор можна зробити невидимим, якщо встановити номер рядка поточної позиції курсору за межами нижньої межі екрану (за допомогою функції 02h). Це найнадійніший спосіб, існують і інші приклади, але далеко не на всіх відеоадаптерах вони дають потрібний результат.

Переривання Int 10h, функція 02h : встановити позицію курсору

Функція позиціонує курсор на екрані. Перед викликом переривання потрібно занести в регістри наступні значення

В AH - код 02 h;

В BH - номер текстової відеосторінки

В DH - номер рядка (I) Y

В DL - номер стовпця (K) X

Примітка : за початок системи координат прийнятий лівий верхній кут екрану, причому вісь X направлена зліва направо, а вісь Y зверху вниз. В стандартному текстовому режимі значення X можна задавати в межах від 0 до 79, а значення Y - від 0 до 25. Якщо задати y=25, то курсор стає невидимим (йде за нижню межу відеосторінки).

Приклад 2.2 Вивести на екран фразу “Графіка” червоними буквами на бірюзовому фоні, використовуючи функції переривання BIOS 10h.

; В сегменті команд

mov AX, data

mov DS, AX

; Виконаємо початкову настройку регістрів

mov cx, 7 ; число символів, що виводяться

mov dl, 36 ; початкова позиція на рядку екрану

```
mov SI, offset mesg ; зсув в рядку тексту
```

```
; Позиціонуємо курсор
```

```
output : mov AH, 02h ; Ф-ція встановки курсору
        mov BH, 0 ; Відеосторінка
        mov DH, 12 ; Рядок
        int 10h ; Переривання BIOS
```

```
; Виведення символу
```

```
mov AH, 09h ; Ф-ція виведення символу
mov AL [SI] ; Символ
mov BL, 34h ; Атрибут
push CX ; Збережемо CX
mov CX, 1 ; Коефіцієнт повторення
int 10h ; Переривання BIOS
pop CX ; Відновлений CX
inc SI ; Зсув на рядок тексту
inc DL ; Зсув по екрану
loop output ; Цикл
```

```
; В сегменті даних
```

```
mesg db 'Графіка'
```

Видно, що виведення кожного окремого символу виконується за допомогою двох програмних блоків: встановки позиції курсору і виведення символів. Оскільки функція 09h не переміщає курсор, це доводиться

робити власноруч перед виведенням кожного наступного символу. З другого боку, використовуючи функції BIOS, ми маємо нагоду виводити кольорові рядки в будь-якому місці екрану.

Переривання Int 10h, функція 03h :

отримати позицію і розмір курсору

Функція визначає поточну позицію курсору на заданій відеосторінці, а також розмір курсору.

Перед викликом переривання вимагається занести в регістри наступні значення :

в AH – код 03h

у BH – номер відеосторінки

Після виконання функції в регістрах будуть розміщені наступні значення :

в CH – номер початкового рядка курсору в знакомісті

в CL – номер кінцевого рядка курсору в знакомісті

в DH – номер рядка поточної позиції курсору

в DL – номер стовпця поточної позиції курсору

Переривання Int 10h, функція 05h :

встановити відеосторінку

Відеосистема комп'ютера представляє можливість використовувати в звичайному текстовому режимі 8 відеосторінок, кожна з яких може незалежно заповнюватися необхідним вмістом. DOS працює тільки з поточною відеосторінкою, для заповнення інших відеосторінок, а також для перемикавання поточної відеосторінки слід використовувати функції переривання BIOS 10h. Методика перемикавання відеосторінок використовуються, наприклад, у відлагоджувачах, які на одній відеосторінці беруть свій інформаційний кадр з текстом відладжуваної програми і інструментами

відлагодження, а іншу відеосторінку використовують для показу “екрану DOS”, тобто виведення на екран результатів роботи відладжуваної програми.

В текстовому режимі дана функція дозволяє вибрати одну з декількох доступних відеосторінок. Перед викликом переривань потрібен занести в регістри наступні значення :

- в AH – код 05h
- в AL - номер відеосторінки, яку потрібно встановити

Приклад 2.3 Перейти на стор. 1

```
mov AH, 05h ; Ф-я перемикання відеосторінки
mov AL, 1   ; Сторінка 1
int 10h
```

Переривання Int 10h, функції 06h і 07h: завдання вікна

Ці функції в заданому місці екрану дисплея створюють кольорові прямокутні вікна заданого розміру. Якщо в створене раніше вікно виведений який-небудь текст, то за допомогою цих же функцій можна прокручувати текст вгору або вниз. При цьому текст, заступивши за край вікна, пропадає, а з-під протилежного краю з'являються порожні рядки із заданими атрибутами кольору. Для заповнення рядків, що з'являються, слід використовувати відповідні функції BIOS, причому контроль місцеположення, довжини і кольору рядків покладаються на програміста. Програми BIOS тільки прокручують задану прямокутну область екрану (разом з текстом в ній).

Перед викликом переривання необхідно занести в регістри наступні значення :

- В AH – код 06h
- В AL – режим створення вікна
- 0 – не прокручувати

Приклад 2.4 Очистити екран і задати атрибути символів за допомогою вікна.

```
mov AH, 06h ; Ф-ія створіння вікна
mov AL, 0   ; Не прокручувати
mov BH, 31h ; Бірюзовий фон, сині символи
mov ch, 0   ; Ліва верхня коор-та Y
mov cl, 0   ; Ліва верхня коор-та X
mov DH, 24  ; Права нижня коор-та Y
mov DL, 79  ; Права нижня коор-та X
int 10h
```

Тут функція 06h, за допомогою якої створюється кольорове вікно розміром у весь екран, використовується в даному прикладі для очищення екрану а також для задання всім знакомістям екрану однакових атрибутів.

Переривання Int 10h, функції 09h, 0Ah: виведення на екран окремих символів і символних рядків

При виклику :

<u>09h</u>	<u>0Ah</u>
AH = 09h	AH = 0Ah
AL = символ	AL = символ
BH = сторінка	BH = сторінка
BL = атрибут (текст.режим) або колір (граф.режим)	CX = коефіцієнт повтор.

CX = коефіцієнт повтор.

Дані функції служать для виведення на екран окремих символів і символьних рядків (в циклі). Функції 09h і 0Ah не виконують фільтрації управляючих символів, тому з їх допомогою можна виводити всі символи кодової сторінки. Передбачено виведення одного і того ж символу задане число раз, що можна використовувати при створенні рамок або орнаментів. Виведення символу не переміщає курсор, тому кожного разу перед застосуванням функції 09h або 0Ah слід позиціонувати курсор за допомогою функції 02h. Різниця функцій 09h і 0Ah полягає в тому, що перша дозволяє вивести символ з будь-яким атрибутом, а друга використовує атрибут тієї позиції, куди виводиться символ.

Приклад 2.5 (продовження попередньої)

```

; встановимо початкові значення в регістрах
    mov DH, 10    ; початковий рядок
    mov DL, 10    ; початковий стовпець
    mov AL, 0     ; початковий символ
; виводимо символи 4-мя рядками по 64 символи
    mov cx, 4     ; 4 рядки (зовнішній цикл)
rows: push cx     ; збережемо лічильник зовнішнього циклу
    mov cx, 64    ; 64 стовпці (внутріш. цикл)
colms: push cx    ; збережемо лічильник внутріш. циклу
; позиціонуємо курсор
    mov AH, 02h   ; Функ. позиціон.
    mov BH, 0     ; сторінка 0
    Int 10h       ;
; виводимо черговий символ
    mov AH, 0Ah   ; Ф-ція виведення символу без задання його
                  ; атрибуту
    mov cx, 1     ; коефіцієнт повторення
    Int 10h       ;
; встановимо нові значення змінних і організуємо цикли
    Inc AL, char   ; наступний символ
    Inc DL         ; наступний стовпець
    pop cx         ; відновлюємо лічильник внутрішнього циклу
    loop colms     ; цикл по стовпцях
    mov DL, 10     ; знову спочатку рядка
    add DH, 2      ; вниз на 2-а рядки
    pop cx         ; відновлений лічильник зовнішнього циклу
    loop rows      ; цикл по рядках

```

Після того, як всім знакомісцям екрану задали однакові атрибути, виведення символів зручно виконувати за допомогою функції 0Ah, яка не вимагає задання атрибута. Перед виведенням чергового символу за допомогою ф-ції 02h встановлюємо позицію курсору.

Переривання Int 10h, функція 0Eh

Необхідно задавати при виклику

AH = 0Eh

AL = символ

BL = колір символу (в граф. реж)

Вона фільтрує управляючі коди (звуковий сигнал, повернення на крок, переклад рядка, повернення каретки), виконуючи відповідні їм дії. Курсор переміщається після виведення кожного символу, що дає можливість виводити цілі рядки. Проте, атрибут символу встановити не можна, символ, що виводиться, отримує колишній атрибут тієї

позиції, куди він виводиться. При необхідності виведення символу з новим атрибутом слід спочатку вивести в задану позицію символ проміжку з необхідним атрибутом (функцією 09h), а потім туди ж послати символ за допомогою функції 0Eh.

Важливою властивістю функції 0Eh є автоматичний перехід на наступний рядок після завершення попередньої, а також прокрутка екрану вгору на один рядок після заповнення самого нижнього рядка.

Переривання Int 10h, функція 0Fh : отримання відеорежиму

Дозволяє отримати поточний відеорежим відеоконтролера

При виклику необхідно задати :

Ан = 0Fh

При поверненні :

АН – число символівних стовпців на екрані.

AL – відеорежим

ВН – активна відео сторінка

Переривання Int 10h, функція 10h підфункція 00h : встановити один реєстр палітри (настройка кольорового режиму)

Ф-ція призначена для 16-кольорових режимів (текстових і графічних). Вона дозволяє перевизначити колір, відповідний одному з кодів кольору . Перед викликом переривання потрібно занести в реєстри наступні значення :

в AX – код 1000h

в BL – номер реєстра палітри (0 –15)

у ВН – колір

Переривання Int 10h, функція 10h, підфункція 01h : встановити колір рамки екрану

Ф-ція дозволяє змінити колір рамки екрану (за замовчанням він чорний). Перед викликом переривання потрібно занести в реєстри наступні значення :

в AX – код 1001h

у ВН – колір

Переривання Int 10h, функція 10h, підфункція 02h: встановити всі реєстри палітри і колір рамки екрану

Ф-ція призначена для 16 – кольорових режимів. Вона дозволяє перевизначити значення 16 реєстрів палітри і реєстра рамки вікна. Перед викликом переривання потрібно занести в реєстри наступні значення :

в AX – код 1002h

в ES:DX – вказівник на масив з 17 байтів (байти 0-15 містять кольори палітри, 16 байт – колір рамки)

Переривання Int 10h, функція 10h, підфункція 03h : переключити біт атрибута мерехтіння / яскравість

Ф-ція переключає значення біта 7-го байта атрибутів символу : мерехтіння або яскравий фон. Перед викликом переривання потрібно занести в реєстри наступні значення :

в AX – код 1003h

в BL – значення біта 7 в байті атрибутів (0 – яскравий фон, 1 – мерехтіння символів)

Функція впливає на режим відображення всіх символів, а не на один конкретний символ. За замовчанням установка біта 7 в байт атрибутів символу викликає його мерехтіння.

**Переривання Int 10h, функція 10h, підфункція 07h :
прочитати один регістр палітри**

Функція призначена для 16-кольорових режимів. Вона дозволяє прочитати значення кольору із заданого регістра палітри. Перед викликом переривання потрібно занести в регістри наступні значення :

в AX – код 1007h

в BL – номер регістра палітри (0 –15)

Після виконання функція повертає значення кольору в регістрі BH.

Приклад 2.6 Прочитати 7-ий регістр палітри

```
mov ax, 1007h      ; функція
mov bl, 7          ; вибираємо 7-ий регістр
Int 10h            ; викликаємо ф-цію BIOS
mov oldcolor, bh   ; зберігаємо значення цього регістра
```

**Переривання Int 10h, функція 10h, підфункція 08h :
прочитати один регістр рамки**

Ф-ція дозволяє прочитати значення кольору рамки екрану з відповідного регістра відеоконтроллера. Перед викликом переривання потрібно занести в регістр AX код 1008h. Після виконання ф-ція повертає значення кольору в регістрі BH.

**Переривання Int 10h, функція 10h, підфункція 09h :
прочитати всі регістри палітри**

Ф-ція призначена для 16-кольорових режимів. Вона дозволяє прочитати і зберегти в оперативній пам'яті значення 16 регістрів палітри і регістра рамки екрану. Перед викликом переривання потрібно занести в регістри наступні значення :

в AX – код 1009h

в ES:DX – вказівник на масив з 17 байтів для збереження кодів кольору.

Після виконання ф-ції байти 0-15 вказаного масиву містять кольори палітри, байт 16-колір рамки.

Приклад 2.7 Прочитати всі значення регістрів палітри і регістра кольору рамки.

```
push ds            ; запам'ятовуємо в стеці значення ds
pop es            ; встановимо ES на сегмент даних
mov ax, 1009h      ; функція
lea dx, old_col_palette ; адресу масиву поміщаємо в dx
Int 10h
Old_col_palette db 17 dup(0) ; значення адреси масиву
```

**Переривання Int 10h, функція 10h, підфункція 10h: встановити один регістр ЦАП
(цифро-аналогового перетворювача)**

Ф-ція призначена для 256-кольорових режимів. Вона дозволяє змінити відтінок, відповідний одному з кодів кольору, шляхом перезапису значень інтенсивності червоного, зеленого і синього кольорів у відповідному регістрі ЦАП. Перед викликом переривання вимагається занести в регістри наступні значення :

- в AX – код 1010h
- у BH – номер регістра ЦАП (0-255)
- в DH - нове значення інтенсивності червоного кольору (0-63)
- в CH – нове значення інтенсивності зеленого кольору (0-63)
- в CL - нове значення інтенсивності синього кольору(0-63)

При роботі з регістром кольору використовуються тільки 6 молодших розрядів кожного байта. Вміст старших розрядів регістрів dh, ch, al BIOS просто ігнорує.

Приклад 2.8 Установка вмісту останнього регістра кольору

```

mov dh, 3Fh      ; інтенсивність червоного кольору;
mov ch, 3Fh      ; інтенсивність зеленого кольору;
mov cl, 3Fh      ; інтенсивність синього кольору;
mov bx, 255      ; номер регістра кольору;
mov ax, 1010h;   код запрошуваної дії;
int 10h          ; виконання запиту;

```

В даному прикладі в останній DAC- регістр відеоконтроллера (0FFh) записується код яскравого білого кольору.

Переривання Int 10h, функція 10h, підфункція 12h :перезавантажити групу регістрів ЦАП

Ф-ція призначена для 256-кольорових регістрів. Вона дозволяє змінити набір відтінків, відповідних групі кольірних кодів, шляхом перезапису значень інтенсивності червоного, зеленого і синього кольорів в послідовно розташованих регістрах ЦАП. Перед викликом переривання потрібно занести в регістри наступні значення :

- в AX – код 1012h
- у BX – номер початкового регістра ЦАП (0-255)
- в CX – число перезавантажуваних регістрів (1-256)
- в ES:DX – вказівник на масив завантажуваних відтінків з 3*CX байт

Масив відтінків повинен містити число 3-байтних рядків, рівне значенню записаному в CX. Перший байт в рядку кодує інтенсивність червоного, другий – зеленого, третій – синього кольорів.

Приклад 2.8 Змінити 63 регистри палітри кольорів, починаючи з 20-м.

```

mov ax, 1012h
mov bx, 20
mov cx, 63
push ds
pop es
lea dx, palette
int 10h
palette db 63*3 dup (0)

```

Переривання Int 10h, функція 10h, підфункція 15h : прочитати один регістр ЦАП

Ф-ція призначена для 256-кольорових режимів. Вона дозволяє прочитати вміст заданого регістра ЦАП.

Перед викликом переривання потрібно занести в регістри наступні значення :

- В AX – код 1015h
- У BX - номер регістра ЦАП (0-255)

Після завершення функції в регістрах будуть розміщені наступні значення :

- в DH - нове значення інтенсивності червоного
- в CH – нове значення інтенсивності зеленого
- в CL - нове значення інтенсивності

**Переривання Int 10h, функція 10h, підфункція 17h :
прочитати групу регістрів ЦАП**

Ф-ція призначена для 256-кольорових режимів. Вона дозволяє зберегти в оперативній пам'яті комп'ютера наборів відтінків, які відповідають групі кольорних кодів.

Перед викликом переривання потрібно занести в регістри наступні значення :

- в AX – код 1017h
- у BX – номер початкового регістра ЦАП (0-255)
- в CX – число зчитуваних регістрів (1-256)
- в ES:DX – вказівник на область пам'яті для збереження зчитаної інформації (масив розміром 3*CX байтів)

Після виконання функції масив відтінків міститиме 3-байтні рядки : перший байт в рядку кодує інтенсивність червоного, другий – зеленого, третій – синього кольорів.

**Переривання Int 10h, функція 11h, підфункція 00h :
завантажити шрифт користувача для текстового відеорежиму**

Ф-ція забезпечує завантаження користувачем заданого шрифту в знакогенератор. Застосовується в текстовому відеорежимі.

Перед викликом функції необхідно занести в регістри наступні значення :

- в AX – код 1100h
- у BH – число байтів в матриці символу
- в BL - номер завантажуваного блоку знакогенератора
- в CX – число завантажуваного символу
- в DX – номер першого завантажуваного символу в таб. кодів
- в ES:BP – вказівник на таблицю, яка містить маски символів завантажуваного шрифту

Весь шифр завантажується цілком (CX = 256, DX = 0), проте, для необхідності можливий перезапис окремої ділянки в наборі символів (в CX записується число замінюваних символів, а в DX - номер першого символу в ділянці, яка замінюється). Пам'ять знакогенератора може містити до 8-ми наборів шрифтів, проте, використовується тільки блок з нульовим номером (BL = 0). В кольоровому текстовому режимі VGA використовується шрифт 8*16, т.е BH = 1 Символи представлені в растровій формі, кожній точці зображення відповідає один біт в масці символу, а кожному рядку - один байт, символ кодується 16 байтами, а повна таблиця шрифту займає 4 Кбайта.

**Переривання Int 10h, функція 11h, підфункція 10h :
завантаження шрифту користувача**

Завантажує таблицю з визначенням шрифту користувача у вказаний блок генератора символів. Перепрограмує контролер на новий розмір шрифту. При використуванні підфункції 10h повинна бути активною відеосторінка 0. Підфункція 00h виконує тіж дії, але не перепрограмує відеоконтроллер.

При виклику :

- в AX – 1110h

у BH – висота символу в числі графічних точок
 в BL - блок генератора
 в CX – число символів, описаних в таблиці
 в DX – код, що призначається першому символу таблиці
 в ES:BP – адреса таблиці

Переривання Int 10h, функція 11h, підфункція 11h : завантаження шрифту ПЗП 8*14

Завантажує шрифт ПЗП BIOS розміром 8*14 графічних точок, що використовується за замовчанням, у вказаний блок генератора символів. При використуванні підфункції 11h повинна бути активною відеосторінка 0. Підфункція 01h виконує такі ж дії, але не перепрограмує відеоконтроллер.

При виклику :

в AX = 1111h
 в BL = блок генератора

Переривання Int 10h, функція 13h: виведення рядка з вказівкою атрибутів як для кожного символу окремо, так і всього рядка

Функція може виконуватися в 4-х варіантах залежно від коду режиму вказуваного в регістрі AL. В режимах 1 і 0 атрибут символів вказується відразу для всього рядка в регістрі BL, причому в режимі 0 курсор не зміщується в процесі виведення, а в режимі 1- зміщується на довжину рядка. В режимах 2 і 3 атрибути символів включаються в рядок, в якому, що виводиться, таким чином чергують коди атрибутів і коди символів, що ускладнює формою рядка, але дозволяє встановлювати атрибути для кожного символу незалежно. Режим 2 відрізняється від режиму 3 тим, що в першому випадку курсор не зміщується, а в другому зміщується на довжину рядка.

При виклику функції 13h в регістрі DX задаються координати початку рядка (в DH- рядок екрану, в DL- стовпець), що виводиться, а регістрі CX- довжина рядка, який в режимах 2 і 3 виявляється за рахунок байтів з атрибутом в два рази більше довжини рядка, який реально з'являється на екрані. Адреса рядка, що виводиться, повинна бути розміщена в регістрах ES:BP. Функція 13h виводить не всі символи, оскільки коди 07h, 08h, 0Ah, 0Dh розглядаються нею як керуючі

При виклику
 AH= 13h
 AL= режим запису
 BH= сторінка
 BL= атрибут (якщо AL= 0 або 1)
 CX= довжина символного рядка
 DH= номер рядка на екрані
 DL= номер стовпця на екрані
 ES: BP = адреса рядка

Висновки: Відеосервіс BIOS забезпечує просте і надійне управління відеосистемою ПК незалежно від її конкретної апаратної реалізації. Проте він має один недолік – порівняно низьку швидкість виконання функцій. З метою підвищення швидкодії відеосистеми використовують альтернативний метод: безпосередня зміна стану регістрів відеоадаптера прикладною програмою. Такий метод є більш швидким і ефективним, проте вимагає від програміста детального знання апаратних особливостей відеоадаптера, що використовується (адрес портів введення-виведення, призначення і структури даних в управляючих регістрах і т.д.)