

Memoria de video VGA

Fuente: Lenguaje ensamblador para computadoras basadas en Intel - KIP R. IRVINE

Contenido

[Memoria de video VGA](#)

[Memoria de video](#)

[Programación de VIDEO con INT 10h](#)

[Funcionamiento del texto de video](#)

[Atributos](#)

[Destello](#)

[Mezcla de colores primarios](#)

[Byte de atributos](#)

[Construcción de los bytes de atributos](#)

[Ejemplo](#)

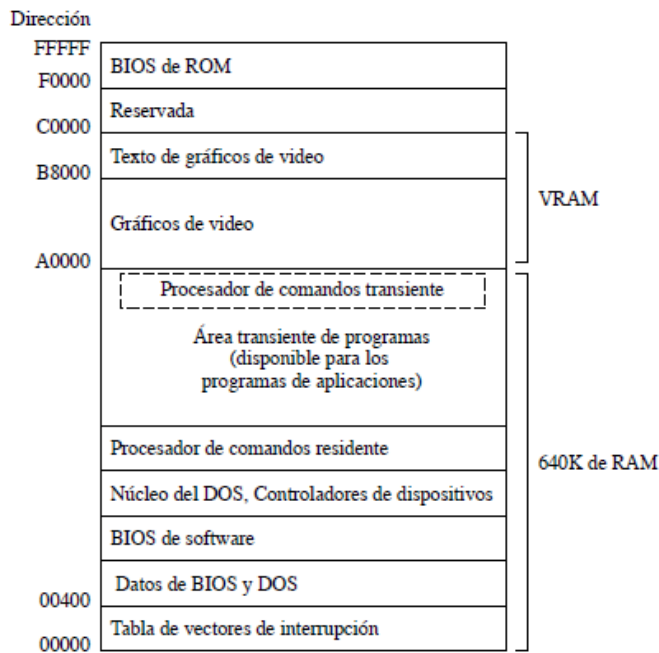
Un monitor de pantalla de cristal líquido (LCD) digital directo recibe un flujo de bits digitales directamente desde el controlador de video, y no requiere del barrido de trama. Por lo general, las pantallas digitales muestran un texto más fino que las pantallas analógicas.

La VRAM almacena datos de video. Tiene doble puerto, y permite que un puerto actualice en forma continua la pantalla, mientras que otro puerto escribe datos.

Memoria de video

El área de la memoria de video (VRAM) en una IBM-PC empieza en la ubicación A0000, la cual se utiliza cuando el adaptador de video cambia al modo de gráficos. Cuando el video se encuentra en modo de texto a color, la ubicación de memoria B8000 almacena todo el texto que se muestra en la pantalla. La pantalla representa un mapa en la memoria, de manera que cada fila y columna en la pantalla corresponde a una palabra de 16 bits en la memoria. Cuando se copia un carácter a la memoria de video, aparece de inmediato en la pantalla.

Mapa de memoria de MS-DOS.



Una interrupción común es INT 10h (Servicios de video). Son procedimientos que muestran rutinas que controlan la posición del cursor, escriben texto a color, desplazan la pantalla y muestran gráficos de video.

Programación de VIDEO con INT 10h

Acceso a nivel de MS-DOS: cualquier computadora que ejecute o emule a MS-DOS puede utilizar la INT 21h para escribir texto en la pantalla de video.

Acceso directo al video: los caracteres se mueven directamente a la RAM de video, por lo que la ejecución es instantánea. Durante la era del MS-DOS, los programas procesadores de palabras y las hojas electrónicas de cálculo utilizaban este método.

Funcionamiento del texto de video

Hay dos modos de vídeo básicos en los sistemas basados en Intel: el modo de texto y el modo de gráficos. Un programa puede ejecutarse en un modo o en el otro, pero no en ambos al mismo tiempo:

- En el modo de texto, los programas escriben caracteres ASCII en la pantalla. El generador de caracteres integrado en el BIOS genera una imagen de mapa de bits

para cada carácter. Un programa no puede dibujar líneas y figuras al azar en el modo de texto.

- En el modo de gráficos, los programas controlan la apariencia de cada píxel de la pantalla. La operación es algo primitiva, ya que no hay funciones integradas para el dibujo de líneas y figuras. En el modo de gráficos podemos utilizar las funciones integradas para escribir texto en la pantalla, y podemos sustituir distintas fuentes por las fuentes integradas. MS Windows proporciona una colección de funciones para dibujar figuras y líneas en el modo de gráficos.

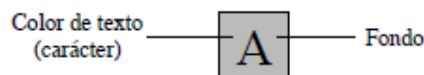
Cuando una computadora inicia en MS-DOS, el controlador de video se establece en el Modo de video 3 (texto a color, 80 columnas por 25 filas de manera predeterminada). En el modo de texto, las filas se enumeran empezando desde la parte superior de la pantalla, fila 0. Cada fila es la altura de una celda de carácter, y utiliza la fuente activa en ese momento. Las columnas se enumeran empezando desde el lado izquierdo de la pantalla, columna 0. Cada columna es la anchura de una celda de carácter.

Páginas de texto de video: La memoria de video en modo de texto se divide en varias páginas de video separadas, cada una de las cuales puede contener una pantalla completa de texto. Los programas pueden mostrar una página mientras escriben texto en las otras páginas ocultas, y pueden cambiar rápidamente de una página a otra. En los días de las aplicaciones MS-DOS de alto rendimiento, a menudo era necesario mantener varias pantallas de texto en memoria al mismo tiempo. Con la popularidad actual de las interfaces gráficas, esta característica de las páginas de texto ya no es importante (la función 05h de INT 10h establece la página de video actual).

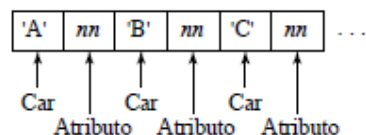
La página de video predeterminada es la página 0.

Atributos

Como se ilustra en los siguientes diagramas, a cada carácter de la pantalla se le asigna un byte de atributo, el cual controla el color del carácter (conocido como color de texto) y el color de la pantalla detrás del carácter (conocido como fondo).



Cada posición en la pantalla de video contiene un solo carácter, junto con su propio atributo (color). El atributo se almacena en un byte separado, después del carácter en la memoria. En la siguiente figura, tres posiciones en la pantalla contienen las letras ABC:

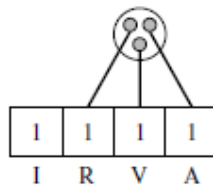


Destello

Los caracteres en la pantalla de video pueden destellar. Para ello, el controlador de video invierte los colores de texto y de fondo de un carácter, a una velocidad predeterminada. De manera predeterminada, cuando una PC inicia en el modo de MS-DOS, el destello está habilitado. Es posible desactivarlo mediante una función del BIOS de video. Además, el destello está desactivado de manera predeterminada cuando se abre una ventana de emulación de MS-DOS estando en MS Windows.

Mezcla de colores primarios

Cada píxel de color en una pantalla de video CRT se genera usando tres rayos de electrones separados: rojo, verde y azul. Un cuarto canal controla la intensidad total, o brillo del píxel. Por lo tanto, todos los colores de texto disponibles se pueden representar mediante valores binarios de 4 bits, en la siguiente forma (I intensidad, R rojo, V verde, A azul). El siguiente diagrama muestra la composición de un píxel blanco:



Al mezclar los tres colores primarios pueden generarse nuevos colores. Además, al encender el bit de intensidad podemos hacer los colores mezclados más brillantes.

Mezcle estos colores primarios...	Para obtener este color	Agregue el bit de intensidad
rojo + verde + azul	gris oscuro	blanco
verde + azul	cyan	cyan claro
rojo + azul	magenta	magenta claro
rojo + verde	café	amarillo
(ningún color)	negro	gris oscuro

Ejemplo de mezcla de colores

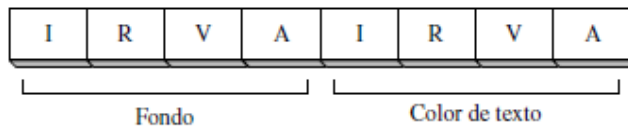
Los colores primarios estilo MS-DOS y los colores mixtos se compilan en una lista de todos los posibles colores de 4 bits, como se muestra en la siguiente tabla. Cada color en la columna de la derecha tiene activado su bit de intensidad.

IRVA	Color	IRVA	Color
0000	negro	1000	gris
0001	azul	1001	azul claro
0010	verde	1010	verde claro
0011	cyan	1011	cyan claro
0100	rojo	1100	rojo claro
0101	magenta	1101	magenta claro
0110	café	1110	amarillo
0111	gris claro	1111	blanco

Codificación de texto a color de cuatro bits.

Byte de atributos

En el modo de texto a color, a cada carácter se le asigna un byte de atributos, el cual consiste en dos códigos de colores de 4 bits: fondo y color de texto:



Destello: Hay una complicación en este esquema simple de colores. Si el adaptador de video tiene habilitado el destello, el bit superior del color de fondo controla el destello del carácter. Cuando este bit se activa, el carácter destella:



Cuando está habilitado el destello, sólo los colores de baja intensidad en la columna izquierda de la tabla anterior están disponibles como colores de fondo (negro, azul, verde, cyan, rojo, magenta, café y gris claro). El color predeterminado cuando se inicia MS-DOS es el 00000111 binario (gris claro sobre fondo negro).

La función 10h de INT 10h tiene varias subfunciones útiles, incluyendo la número 03h, que permite que el bit superior de un atributo de color controle la intensidad o el destello del carácter. En la siguiente tabla podrá consultar los detalles:

Función 10h de INT 10h, subfunción 03h	
Descripción	Establece el modo de destello e intensidad
Recibe	AH = 10h AL = 3 BL = modo de destello (0 = habilita intensidad, 1 = habilita destello)
Devuelve	Nada
Llamada de ejemplo	<pre> mov ah,10h mov al,3 mov bl,1 ; habilita el destello int 10h </pre>
Notas	Cambia el texto en la pantalla, entre el modo de destello y el modo de alta intensidad. En MS Windows, el destello sólo puede ocurrir cuando la aplicación se ejecuta en modo de pantalla completa

Construcción de los bytes de atributos

Para construir un byte de atributos de video a partir de dos colores (color de texto y de fondo), se puede asignar directamente o usar el operador SHL del ensamblador para desplazar los bits del color de fondo cuatro posiciones a la izquierda, y aplique un OR entre estos bits y el color del texto. Por ejemplo, las siguientes instrucciones crean un atributo de texto gris claro sobre un fondo azul:

```

azul = 1
grisClaro = 1111b
mov bh,(azul SHL 4) OR grisClaro          ; 00010111

```

0	0	0	1	0	1	1	1
FONDO				TEXTO			

Las siguientes instrucciones crean caracteres blancos sobre un fondo rojo:

```

blanco = 1111b
rojo = 100b
mov bh,(rojo SHL 4) OR blanco          ; 01001111

```

0	1	0	0	1	1	1	1
FONDO				TEXTO			

Las siguientes líneas producen letras azules sobre un fondo café:

```
azul = 1  
cafe = 110b  
mov bh,((cafe SHL 4) OR azul)          ; 01100001
```

0	1	1	0	0	0	0	1
FONDO				TEXTO			

Las fuentes y los colores pueden aparecer un poco distintos al ejecutar el mismo programa en diferentes sistemas operativos. Por ejemplo, en Windows 2000 y XP el destello está deshabilitado, a menos que cambiemos a modo de pantalla completa. Lo mismo se aplica para la visualización de gráficos mediante INT 10h.

Ejemplo

Este ejemplo imprime "Organización del Computador" escribiendo directamente en la memoria de vídeo. En la memoria VGA VRAM: el primer byte es un carácter ASCII, el siguiente byte es un atributo de carácter. Si cambiamos el segundo byte, puede cambiar el color del carácter incluso después de imprimirlo. El atributo de carácter es un valor de 8 bits. Los 4 bits más significativos (MSB) establecen el color de fondo y los 4 bits menos significativos (LSB) establecen el color de primer plano.

```
name "orga"
```

```
name "orga"
```

```
; Este ejemplo imprime "Organización del Computador"  
; escribiendo directamente en la memoria de video.  
; En la memoria VGA VRAM: el primer byte es un caracter ASCII,  
; el siguiente byte es un atributo de caracter.  
; Si cambiamos el segundo byte, puede cambiar el color del caracter  
; incluso después de imprimirlo. El atributo de caracter es un valor de 8 bits.  
; Los 4 bits más significativos (MSB) establecen el color de fondo y  
; los 4 bits menos significativos (LSB) establecen el color de primer plano.
```

; hex	bin	color
; 0	0000	negro
; 1	0001	azul
; 2	0010	verde
; 3	0011	cyan
; 4	0100	rojo
; 5	0101	magenta
; 6	0110	marrón
; 7	0111	gris claro
; 8	1000	gris oscuro
; 9	1001	azul claro
; a	1010	verde claro
; b	1011	cyan claro
; c	1100	rojo claro
; d	1101	magenta claro
; e	1110	amarillo
; f	1111	blanco

org 100h

; Modo de video 3 (texto a color, 80 columnas por 25 filas de manera predeterminada)

mov ax, 3 ; mode texto 80x25, 16 colores, 8 páginas (ax = 3 significa ah=0, al=3)

int 10h ; ejecuta!

; Cancela el destello y establece los colores:

mov ax, 1003h ; al = 03 permite que el MSB controle la intensidad del color del caracter

mov bx, 0 ; Cancela el destello

int 10h

; Define el registro de segmento en la VRAM para texto a color (B8000):

mov ax, 0b800h

mov ds, ax

; imprime "Organización del Computador"

; El primer byte es un código ASCII ([02h] es 0), el segundo el código de color([03h]).

mov [02h], '0'

mov [04h], 'r'

mov [06h], 'g'

mov [08h], 'a'


```
mov [0ah], 'n'  
mov [0ch], 'i'  
mov [0eh], 'z'  
mov [10h], 'a'  
mov [12h], 'c'  
mov [14h], 'i'  
mov [16h], 'o'  
mov [18h], 'n'  
mov [1ah], ''  
mov [1ch], 'd'  
mov [1eh], 'e'  
mov [20h], 'l'  
mov [22h], ''  
mov [24h], 'C'  
mov [26h], 'o'  
mov [28h], 'm'  
mov [2ah], 'p'  
mov [2ch], 'u'  
mov [2eh], 't'  
mov [30h], 'a'  
mov [32h], 'd'  
mov [34h], 'o'  
mov [36h], 'r'
```

; colorea todos los caracteres:

```
mov cx, 27 ; número de caracteres.
```

```
mov di, 03h ; Comienza desde el byte siguiente a 'O'
```

```
c: mov [di], 11101100b ; rojo claro(1100) sobre amarillo(1110)
```

```
    add di, 2 ; busca el siguiente código ascii en la memoria vga.
```

```
    loop c
```

; espera que se apriete alguna tecla para devolver el control al sistema operativo:

```
mov ah, 0
```

```
int 16h
```

```
ret
```

