ECE291

Lecture 11

Text mode video

Lecture outline

- Comments on the exam
- Video display hardware
- Text video mode
- Controlling and accessing video display using DOS and BIOS functions
- Accessing the video memory directly

Video output

- Video is the primary form of communication between a computer and a human.
- The monochrome (single color) monitor uses one wire for video data, one for horizontal sync, and one for vertical sync.
- A color video monitor uses three video signals: **red**, **green**, & **blue**
  - these monitors are called RGB monitors and convert the analog RGB signals to an optical image.
- The RGB monitor is available as either an analog or TTL (digital) monitor.

TTL RGB monitor

- Uses TTL level signals (0 or 5V) as video inputs (RGB) and an extra signal called intensity to allow change in intensity
- Used in the CGA (Color Graphics adaptor) system found in older computers
- Can display a total of 16 different colors (eight are generated at high intensity and eight at low intensity)
- Example:

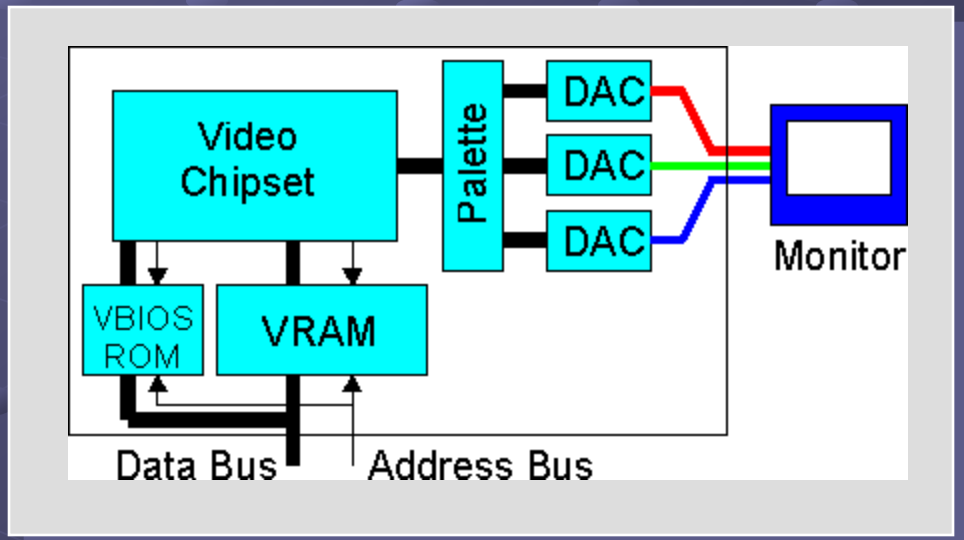
Intensity	Red	Green	Blue	Color
0	0	0	0	Black
1	0	0	0	Gray
1	0	1	0	Light green
1	1	0	0	Light red
1	1	1	1	Bright white

The analog RGB monitor

- Uses analog signals - any voltage between 0.0 V and 0.7 V
- This allow an infinite number of colors to be displayed
- In practice a finite number of levels is generated (16K, 256K, 16M, colors depending on the standard)
- Analog displays use a **digital-to-analog converter** (DAC) to generate each color video voltage
- A common standard uses a 6-bit DAC to generate 64 different video levels between 0 V and 0.7 V
  - this allows 64x64x64 colors to be displayed, or 262,144 (256 K) colors
- 8-bit DACs allow 256x256x256 or 16M colors

The analog RGB monitor

- The Video Adapter converts digital information from the CPU to analog signals for the monitor.
- VRAM/DRAM Video/Dynamic Random Access Memory
  - Stores screen content
- Video BIOS
  - Stores character mappings
- Palette registers
  - Defines R/G/B color values
- Graphic accelerator
  - Hardware implemented graphic routines
- DAC
  - Generates analog Red/Green/Blue signals



The analog RGB monitor

- Example: video generation used in video standards such as EGA (enhanced graphic adapter) and VGA (variable graphics array)
- A high-speed palette SRAM (access time less than 40ns) stores 256 different codes that represent 256 different hues (18-bit codes)
- This 18-bit code is applied to the DACs
- The SRAM is addressed by 8-bit code that is stored in the computer VRAM to specify color of the pixel
- Once the color code is selected, the three DACs convert it to three video voltages for the monitor to display a picture element (pixel)

The Analog RGB Monitor Example of Video Generation (cont.)

- Any change in the color codes is accomplished during retrace (moving the electron beam to the upper left-hand corner for vertical retrace and to the left margin of the screen for horizontal retrace)
- The resolution and color depth of the display (e.g., 640x400) determines the amount of memory required by the video interface card
- 640x400 resolution with 256 colors (8 bits per pixel) 256K bytes of memory are required to store all the pixels for the display

Text mode video

- There is not a single common device for supporting video displays
- There are numerous display adapter cards available for the PC
- Each supports several different display modes
- We'll discuss the 80x25 text display mode which is supported by most of display adapters
- The 80x25 text display is a two dimensional array of words with each word in the array corresponding a character on the screen
- Storing the data into this array affects the characters appearing on the display

Text mode video

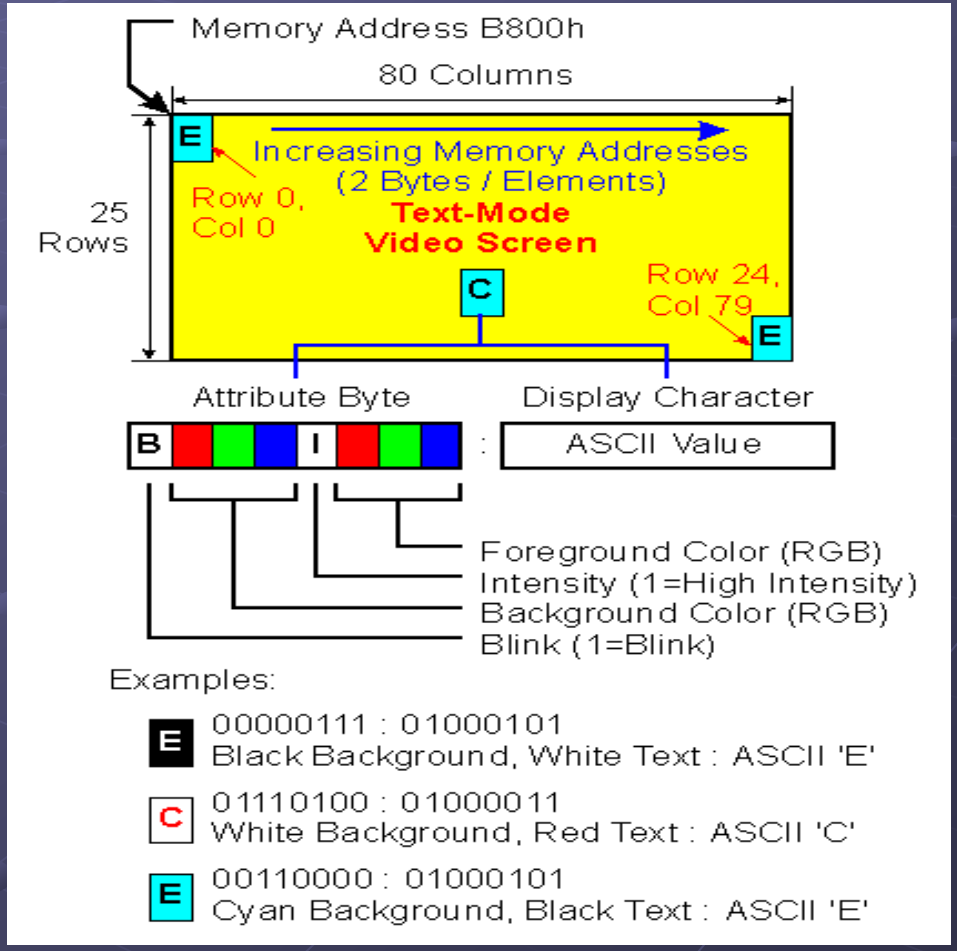
- Each text page occupies under 4K bytes of memory
- 80(columns) x 25 (rows) x 2 (bytes) = 4000 bytes
- The LO byte contains the ASCII code of the character to display
- The HO byte contains the attribute byte
- Display adapters provide 32 K for text displays and let you select one of eight different pages
- Each display begins on a 4K boundary, at address:
  - B800:0000, B800:1000, B800:2000, …. B800:7000

Text mode video

The attribute byte controls underlying background and foreground colors, intensity and blinking video

Choose your colors with care (some combinations of foreground and background colors are not readable)

Do not overdo blinking text on the screen



The cursor

- A pointer to the “insertion point” on the screen
- When you use DOS/BIOS functions to display a character, it displays where the cursor points
- The cursor then moves to the next column
- Other functions let you move backwards or up/down

DOS/BIOS functions revisited

- Recall INT 21h functions
  - 02h, 06h output characters to screen at current cursor position
  - 09h output '$' terminated string to screen beginning at current cursor position
- Recall INT 10h functions
  - 02h sets cursor position (including page)
  - 03h reads cursor position (including page)
  - 05h set active display page

Writing characters directly

- Since the VRAM is memory mapped, you can use MOV instructions to write data directly to the display
- Typically, we set the ES register to B800h so that the extra segment can be used to address the VRAM
- Now video display can be accessed just like a 2D word array

Example

- Calculate the offset from the beginning of the VRAM segment (B8000h) for an arbitrary page (P), row (Y) and column (X) in an 80x25 text display mode
  - Offset = 1000h * page + 160 * Y + 2*X

String instructions

- Idea: Setup a data transfer and go
- Do an operation on source [DS:SI] and destination [ES:DI] and change SI and DI depending on the direction flag
- Transfer data much more quickly than loops and movs
- Think of the following instructions in terms of their equivalents
  - You can't do memory to memory operations with other opcodes.
  - The add's at the end of the equivalent code don't affect the flags.

String instructions

- MOVS – Move source to destination

```
Mov byte [es:di], byte [ds:si]
add si, 1  ; if CLD
add di, 1
```

- CMPS – Compare source to destination and set ZF if [ES:DI] = [DS:SI]

```
cmp byte [es:di], byte [ds:si]
add si, 1  ; If CLD
add di, 1
```

String instructions

- STOS – Store AL/AX/EAX into destination

```
mov byte [es:di], al
add di, 1  ; If CLD
```

- LODS – Load destination into AL/AX/EAX

```
mov al, byte [ds:si]
add si, 1  ; If CLD
```

- SCAS – Compare destination to AL… and set ZF if [ES:DI] = AL…

```
cmp byte [es:di], al
add di, 1  ; If CLD
```

String instructions

- Each of the instructions should be appended with B, W or D for byte, word, or double word sized transfers.
- REP – What makes all this useful
  - This is a prefix to the above opcodes
  - REP/REPE/REPZ
    - DEC CX
    - LOOP until CX = 0 while ZF = 1
  - REPNE/REPNZ
    - DEC CX
    - Loop until CX = 0 while ZF = 0

String instructions

```
; Example: Copying a display buffer to the screen
CLD                         ; clear dir flag so we go up

; setup source
MOV SI, DisplayBuffer       ; offset with respect to DS

; setup destination
MOV AX, VidGrSeg            ; B800
MOV ES, AX                  ; set destination segment as ES
MOV DI, 0                   ; start of screen

; setup counter
MOV CX, (320*200 / 4)       ; moving 4 bytes at a time
REP MOVSD                   ; this takes awhile
```