# DOS and BIOS Interrupts

- **DOS and BIOS interrupts** are used to perform some very useful functions, such as **displaying data to the monitor, reading data from keyboard**, etc.

- They are **used by identifying the interrupt option type**, which is the value stored in register **AH** and providing, whatever extra information that the specific option requires.

# BIOS Interrupt 10H

- **Option 0H – Sets video mode.**

- **Registers used:**
  - ☐ **AH = 0H**
  - ☐ **AL = Video Mode.**
    - ▪ **3H - CGA Color text** of 80X25
    - ▪ **7H - Monochrome text** of 80X25

- **Ex:**
  - ☐ **MOV AH,0**
  - ☐ **MOV AL,7**
  - ☐ **INT 10H**

# BIOS Interrupt 10H

- **Option 2H – Sets the cursor to a specific location.**

- **Registers used:**
  - **AH = 2H**
  - **BH = 0H selects Page 0.**
  - **DH = Row position.**
  - **DL = Column position.**

# BIOS Interrupt 10H

- **Ex:**
  - MOV AH,2
  - MOV BH,0
  - MOV DH,12
  - MOV DL,39
  - INT 10H

# BIOS Interrupt 10H

- **Option 6H – Scroll window up. This interrupt is also used to clear the screen when you set AL = 0.**

- **Registers used:**
  - **AH = 6H**
  - **AL = number of lines to scroll.**
  - **BH = display attribute.**
  - **CH = y coordinate of top left.**
  - **CL = x coordinate of top left.**
  - **DH = y coordinate of lower right.**
  - **DL = x coordinate of lower right.**

# BIOS Interrupt 10H

- **Clear Screen Example:**
  - □ MOV AH,6
  - □ MOV AL,0
  - □ MOV BH,7
  - □ MOV CH,0
  - □ MOV CL,0
  - □ MOV DH,24
  - □ MOV DL,79
  - □ INT 10H
- **The code above may be shortened by using AX, BX and DX registers to move word size data instead of byte size data.**

# BIOS Interrupt 10H

- **Option 7H – Scroll window down. This interrupt is also used to clear the screen when you set AL = 0.**

- **Registers used:**
  - **AH = 7H**
  - **AL = number of lines to scroll.**
  - **BH = display attribute.**
  - **CH = y coordinate of top left.**
  - **CL = x coordinate of top left.**
  - **DH = y coordinate of lower right.**
  - **DL = x coordinate of lower right.**

# BIOS Interrupt 10H

- **Option 8H – Read a character and its attribute at the cursor position.**

- **Registers used:**
  - ☐ **AH = 8H and returned attribute value.**
  - ☐ **AL = Returned ASCII value.**
  - ☐ **BH = display page.**

# BIOS Interrupt 10H

- **Option 9H – Write a character and its attribute at the cursor position.**

- **Registers used:**
  - **AH = 9H.**
  - **AL = ASCII value.**
  - **BH = display page.**
  - **BL = attribute.**
  - **CX = number of characters to write.**

# Attribute Definition

| Blinking | Background | | | Intensity | Foreground | | |
|:---:|:---:|:---:|:---:|:---:|:---:|:---:|:---:|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

- **Monochrome display attributes**
  - **Blinking**
    - **D7 = 0 - Non-blinking**
    - **D7 = 1 - Blinking**
  - **Intensity**
    - **D3=0 - Normal intensity**
    - **D3=1 - Highlighted intensity**
  - **Background and foreground**
    - **D6 D5 D4 and D2 D1 D0**
      - **White = 0 0 0**
      - **Black = 1 1 1**

# Attribute Definition

| Blinking | Background | | | Intensity | Foreground | | |
|---|---|---|---|---|---|---|---|
| | R | G | B | | R | G | B |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

- **Color display attributes**
  - **Blinking**
    - **D7 = 0 - Non-blinking**
    - **D7 = 1 - Blinking**
  - **Intensity**
    - **D3=0 - Normal intensity**
    - **D3=1 - Highlighted intensity**
  - **Background and foreground**
    - **D6 D5 D4 and D2 D1 D0**
      - **RGB values defined by the table to the right.**

| I | R | G | B | Color |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Black |
| 0 | 0 | 0 | 1 | Blue |
| 0 | 0 | 1 | 0 | Green |
| 0 | 0 | 1 | 1 | Cyan |
| 0 | 1 | 0 | 0 | Red |
| 0 | 1 | 0 | 1 | Magenta |
| 0 | 1 | 1 | 0 | Brown |
| 0 | 1 | 1 | 1 | White |
| 1 | 0 | 0 | 0 | Gray |
| 1 | 0 | 0 | 1 | Light blue |
| 1 | 0 | 1 | 0 | Light green |
| 1 | 0 | 1 | 1 | Light cyan |
| 1 | 1 | 0 | 0 | Light red |
| 1 | 1 | 0 | 1 | Light magenta |
| 1 | 1 | 1 | 0 | Yellow |
| 1 | 1 | 1 | 1 | High intensity white |

# DOS Interrupt 21H

- **Option 1 – Inputs a single character from keyboard and echoes it to the monitor.**

- **Registers used:**
  - ☐ **AH = 1**
  - ☐ **AL = the character inputted from keyboard.**

- **Ex:**
  - ☐ **MOV AH,1**
  - ☐ **INT 21H**

# DOS Interrupt 21H

- **Option 2 –** <mark>**Outputs a single character to the monitor.**</mark>

- **Registers used:**
  - **AH = 2**
  - **DL = the character to be displayed.**

- **Ex:**
  - **MOV AH,2**
  - **MOV DL,'A'**
  - **INT 21H**

# DOS Interrupt 21H

- **Option 9 – Outputs a string of data, terminated by a $ to the monitor.**

- **Registers used:**
  - **AH = 9**
  - **DX = the offset address of the data to be displayed.**

- **Ex:**
  - **MOV AH,09**
  - **MOV DX,OFFSET MESS1**
  - **INT 21H**

# DOS Interrupt 21H

- **Option 0AH – Inputs a string of data from the keyboard.**

- **Registers used:**
  - **AH = 9**
  - **DX = the offset address of the location where string will be stored.**

- **DOS requires that a buffer be defined in the data segment. It should be defined as follows:**

  - **1$^{st}$ byte contains the size of the buffer.**
  - **2$^{nd}$ byte is used by DOS to store the number of bytes stored.**

# DOS Interrupt 21H

- **Ex:**
  - .DATA
  - BUFFER1      DB      15,?,15 DUP (FF)
  - .
  - .
  - MOV AH,0AH
  - MOV DX,OFFSET BUFFER1
  - INT 21H
- **Assume "Go Tigers!" was entered on the keyboard.**
  - BUFFER1 = 10,10,'Go Tigers!',CR,255,255,255,255

# DOS Interrupt 21H

- **Option 4CH – Terminates a process, by returning control to a parent process or to DOS.**

- **Registers used:**
  - ☐ **AH = 4CH**
  - ☐ **AL = binary return code.**

- **Ex:**
  - ☐ **MOV AH,4CH**
  - ☐ **INT 21H**