# Laboratory Exercise #3-4
# BIOS Interrupts

**Target Course Outcome:**
**CO2:** Designs a microprocessor-based firmware integrating a microprocessor with supporting peripherals and devices.

**Objectives:**
To understand the concept of subroutines and use it in assembly language programs.
To know the different DOS interrupts and use such interrupts in assembly language programs.

**Tools Required:**
Emu8086 Emulator

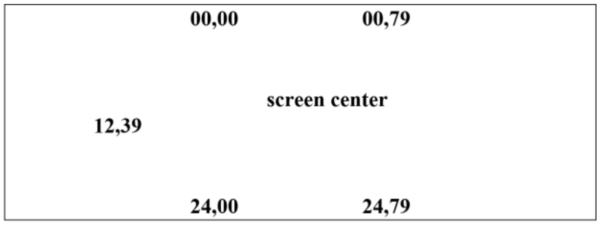
## Part 1:

### Basic Input/Output System (BIOS)

Included within the BIOS one will find a set of interrupts that are extremely useful. These interrupt subroutines may be used by issuing a software interrupt call (**INT** ***type***). The user will also have to identify which function is being used by setting the AH register to a specific value. Other registers may also need to be modified. One should note that interrupts are a FAR call, in other words, when an interrupt is executed the return address is stored as CS: IP.

### BIOS INT 10H

INT 10H are stored in the BIOS ROM of the IBM PC type computers, and they are used to control the screen video. The monitor screen in normal text mode is composed of 25 rows and 80 columns, and text mode is the default mode whenever a monitor is turned on. There are several types of monitors including: MDA, MCGA, CGA, EGA and VGA. In all these modes the text screen is 80X25 characters long. The text locations are numbered from 0 to 24 for the rows and 0 to 79 for the columns as shown in the diagram below.



Several functions are performed by INT 10H, therefore the programmer needs to identify which one is being used by storing an appropriate value in register AH.

For example:

```
AH = 00H          ;Selects the change video mode function
INT 10H           ;Executes BIOS interrupt 10H.
```

Depending on the function being used, other register may be used to pass information to the interrupt subroutine. Following are the descriptions of several of the most common INT 10H functions.

**INT 10H Function 00H: Change video mode**

AH = 00H
AL = Video Mode
03H – 80X25 CGA text
07H – 80X25 Monochrome text

Code example to set video mode to 80X25 CGA text

```
MOV AL,03H
MOV AH,00H
INT 10H
```

INT 10H Function 02H: Set Cursor Position

AH = 02H
BH = Page number DH = Row number
DL = Column number

BH = 0 when graphics mode is in use.

INT 10H Function 06H: Scroll window up INT 10H Function 07H: Scroll window down

AH = 06H to scroll up or 07H to scroll down
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

If AL = 0 the entire window is blank, otherwise, the screen will be scrolled upward/downward by the value stored in AL. Lines scrolling off the screen are lost and blank lines are scrolled in at the bottom/top according to the attribute in BH.

INT 10H Function 08H: Read character and attribute at cursor position

AH = 08H
BH = Display page

AH = Returned attribute byte
AL = Returned ASCII character code

INT 10H Function 09H: Write character and attribute at cursor position

AH = 09H
AL = ASCII character code
BH = Display page
BL = Attribute
CX = Number of characters to write

The character attribute is defined as shown in the following tables:

**Monochrome display attributes**

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

D7 non-blinking= 0
Blinking = 1

D3 Normal intensity = 0
Highlighted intensity = 1

D6 D5 D4 and D2 D1 D0 White = 0 0 0
Black = 1 1 1

**CGA display attributes**

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

D7 non-blinking= 0
Blinking = 1

D3 Normal intensity = 0
Highlighted intensity = 1

Both blinking and intensity are applied to foreground only
D6 D5 D4 and D2 D1 D0 Color as defined on the following table

**Color Attributes**

| 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 |

**Part II**

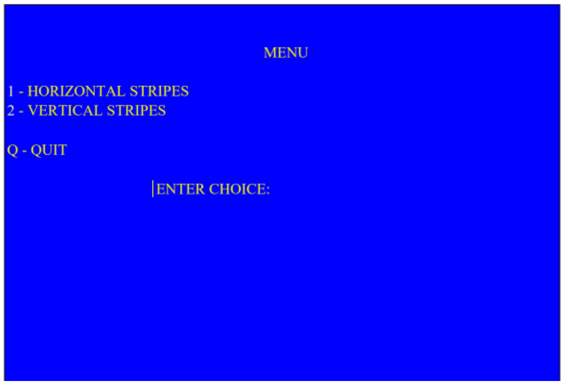
Use of BIOS interrupts in accessing the screen/monitor

**Activity #1**

Using the interrupts described above, write a program to:

1 - Clear the screen.

2 - Create the following menu of choices as shown below



```
                         MENU

1 - HORIZONTAL STRIPES
2 - VERTICAL STRIPES

Q - QUIT

             |ENTER CHOICE:
```

The background color is blue and the foreground color for the letters is yellow.

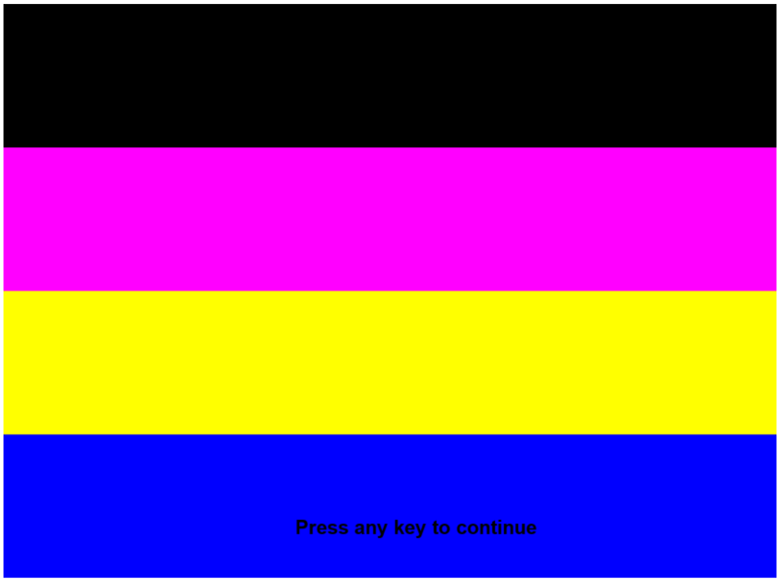
3 – For each choice generate the chosen pattern. Make sure the patterns have four differently colored stripes and the message: “Press any key to continue.” , is displayed. Display the new screen until any key is pressed on the keyboard then return to the main screen to display the menu of choices again.

Include or create subroutines in creating the program. The subroutines are identified below.

CLEAR_SCREEN – this subroutine will clear the screen.

DISP_MESS – this subroutine displays messages to the screen. Use register DX to pass the initial address of the message to displayed to the subroutine.

Below is an example with 4 horizontal stripes.



**Activity #2**

Write a program that will ask the user to input a message. Displaythe inputted text starting at the center of the screen (row 12). Repeat the text display until the last row of the screen. Text must be colored blue and blinking. Label the filename as Exp3-4.asm

## Copyright Information

Author : Rosana J. Ferolin (rjferolin@usc.edu.ph)
Date of release : August 7, 2020
Version : 1.0

## Change log:

| Date | Version | Author | Changes |
|---|---|---|---|
| Aug. 7, 2020 | 1.0 | Rosana J. Ferolin | Initial Draft |
| Sep. 3, 2025 | 2.0 | Marlowe Edgar C. Burce | Revised activity and procedures |