

Microprocessor

(Unit-3 :: Lectures – 17/18)

(BEI – I/II & BCT – II/II)

Er. Dinesh Baniya Kshatri
Assistant Professor
Institute of Engineering, Thapathali Campus

BIOS Interrupt – (INT 10H)

(Function 00H) – [1]

- The function number (00H) must be stored in AH register
- It sets the video mode of the screen
- Video mode must be stored in AL register
- Examples of video modes are given below:

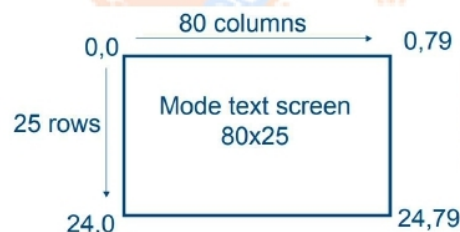
AL = 1	40x25	Text	16 colours
AL = 3	80x25	Text	16 colours

Er. Dinesh Baniya Kshatri

2

BIOS Interrupt – (INT 10H)

(Function 00H) – [2]



In code segment

```
mov ah, 00h
mov al, 03h
int 10h
```

Er. Dinesh Baniya Kshatri

3

BIOS Interrupt – (INT 10H)

(Function 02H) – [1]

- The function number (02H) must be stored in AH register
- Positions cursor at specified coordinates in text mode screen
- Additional parameters required by the function include:
 - DH = Row Number (0...24)
 - DL = Column Number (0...39 or 0...79 based on video mode)
 - BH = Page Number where cursor is to be placed

Er. Dinesh Baniya Kshatri

4

BIOS Interrupt – (INT 10H) (Function 02H) – [2]

- Page Numbering: (0 is the active / default page number)
 - 0 through 7 for 40 column mode
 - 0 through 3 for 80 column mode

In code segment:

```
mov ah, 02h
mov dh, 12    ;Row12
mov dl, 20    ;Column 20
mov bh, 00h   ;Page 0
int 10h
```

Er. Dinesh Baniya Kshatri 5

BIOS Interrupt – (INT 10H) (Functions 06H / 07H) – [1]

- The function number (06H) or (07H) must be stored in AH register
- Causes a window to scroll up (06H) or scroll down (07H)
- Additional parameters required by the function include:
 - AL = number of lines to scroll (Screen is cleared if AL = 0)
 - CH = Row coordinate of upper left corner of window
 - CL = Column coordinate of upper left corner of window
 - DH = Row coordinate of lower right corner of window
 - DL = Column coordinate of lower right corner of window
 - BH = Attribute to use on scrolled region

Er. Dinesh Baniya Kshatri 6

BIOS Interrupt – (INT 10H) (Functions 06H / 07H) – [2]

- Attribute value is an 8-bit number
 - Lower 4-bits represent foreground color
 - Upper 4-bits represent background color

Bit	7	6	5	4	3	2	1	0
	P		Background				I	Foreground

- P = 1 for blinking characters, 0 = non-blinking characters
- I = 1 for intense foreground characters, 0 = normal foreground characters

Er. Dinesh Baniya Kshatri 7

BIOS Interrupt – (INT 10H) (Functions 06H / 07H) – [3]

- Possibilities for background and foreground colors are as follows:

Bit	2	1	0	
	6	5	4	
0	0	0	0	Black
0	0	0	1	Blue
0	1	0	0	Green
0	1	1	0	Cyan
1	0	0	0	Red
1	0	1	0	Magenta
1	1	0	0	Brown
1	1	1	1	Light gray

Er. Dinesh Baniya Kshatri 8

BIOS Interrupt – (INT 10H) (Functions 06H / 07H) – [4]

In code segment

```

mov ah, 06 h
mov ch, 10      ;Upper left row corner 10
mov cl, 20      ;Upper left column corner 20
mov dh,12       ;Lower right row corner 12
mov dl,22       ;Lower right column corner 22
mov bh, 07 h    ;White on black attribute
mov al,03 h     ;Number of lines to be scrolled
int 10h

```

Er. Dinesh Baniya Kshatri 9

BIOS Interrupt – (INT 10H) (Functions 09H) – [1]

- The function number (09H) must be stored in AH register
- Writes attribute and character on current cursor position once or many times
- Additional parameters required by the function include:
 - AL = ASCII code of character to be displayed
 - BH = Page number
 - BL = Attribute
 - CX = Number of times (repetition factor)

Er. Dinesh Baniya Kshatri 10

BIOS Interrupt – (INT 10H) (Functions 09H) – [2]

In code segment

```

mov ah, 09 h
mov bh, 00 h    ;Page0
mov bl,07 h     ;White on black attribute
mov cx,0004 h  ;Display four times
mov al,'A'      ;character 'A'
int 10h

```

Er. Dinesh Baniya Kshatri 11

BIOS Interrupt – (INT 10H) (Example – 1)

- Write code to set the cursor at column 35, row 7.

```

MOV AH,02H
MOV BH,00H     ;set page 0
MOV DL,35      ;column in DL
MOV DH,07      ;row in DH
INT 10H

```

Er. Dinesh Baniya Kshatri 12

BIOS Interrupt – (INT 10H) (Example – 2)

- Write code to clear entire screen with white background and red foreground (Assume video mode set at 80 × 25)

```

MOV AX,0600h ;AH=06h & AL=00h
MOV BH,74h ;White background (7)
           red foreground (4)
MOV CX,0000h ;row 0 col 0
MOV DX,184Fh ;row 24 col 79 (in Hex)
INT 10h

```

Er. Dinesh Baniya Kshatri 13

BIOS Interrupt – (INT 10H) (Example – 3)

- Write code to scroll up 3 lines in window [(12,25), (18, 54)] with white background and red foreground

```

MOV AX,0603h ;scroll up 3 lines
MOV BH,74 ;White background red
           foreground
MOV CX,0C19h ;From row 12,Column 25
MOV DX,1236h ;to row 18,column 54
INT 10h

```

Er. Dinesh Baniya Kshatri 14

Sample 8086 Program – [1]

- This program displays the character 'D' on the screen

.MODEL SMALL	INT 21H
.STACK 64	MOV AH, 4CH
.CODE	INT 21H
MOV DL, 'D'	END
MOV AH, 02H	

Er. Dinesh Baniya Kshatri 15

Carriage Return and Line Feed

- Carriage Return:**
 - ASCII value = 0DH or 13D
 - It is a control character in the ASCII code
 - It moves position of the cursor to the beginning on the same line
- Line Feed:**
 - ASCII value = 0AH or 10D
 - It is also a control character in the ASCII code
 - It signifies the end of a line of text and starts a new line

Er. Dinesh Baniya Kshatri 16

Sample 8086 Program – [2]

- This program displays **Hello World** on the screen

.MODEL SMALL	MOV AH,09H
.STACK 64	MOV DX, OFFSET MSG
.DATA	INT 21H
MSG DB 'Hello World', 0DH, 0AH, '\$'	MOV AH, 4CH
.CODE	INT 21H
MOV AX, @DATA	END
MOV DS, AX	

Er. Dinesh Baniya Kshatri

17

Sample 8086 Program – [3]

- The following program allows a user to enter 20 characters from the keyboard, and stores the characters entered into a buffer. The program terminates after the Return key is pressed.

.MODEL SMALL	MOV DS, AX	CMP AL, 0DH
.STACK 64	LEA SI, buffer	JNE AGAIN
.DATA	MOV AH, 01H	MOV AH, 4CH
buffer DB 20 DUP(?)	AGAIN: INT 21H	INT 21H
.CODE	MOV [SI], AL	END
MOV AX,@DATA	INC SI	

Er. Dinesh Baniya Kshatri

18

Sample 8086 Program – [4]

- Write a program to calculate the total sum of five words of data. The decimal data is as follows: 27345, 28521, 29533, 30105, 32375

.MODEL SMALL	MOV CX, COUNT	JNZ BACK
.STACK 64	MOV SI, OFFSET ARRAY	MOV SUM, AX
.DATA	MOV AX, 0000H	MOV SUM+2,BX
COUNT EQU 0005H	MOV BX, AX	MOV AH, 4CH
ARRAY DW 27345, 28521, 29533, 30105, 32375	BACK: ADD AX, [SI]	INT 21H
SUM DW 2 DUP(?)	ADC BX, 0	END
.CODE	INC SI	
MOV AX, @DATA	INC SI	
MOV DS, AX	DEC CX	

Er. Dinesh Baniya Kshatri

19

Sample 8086 Program – [5]

- Two bytes stored in the data segment are multiplied. The result is stored in a memory location called PROD.

.MODEL SMALL	.CODE	MOV AH, 4CH
.STACK 64	MOV AX,@DATA	INT 21H
.DATA	MOV DS, AX	END
MULT DB 0AH	MOV AL, MULT	
MULP DB 0F6H	MUL MULT	
PROD DW ?	MOV PROD, AX	

Er. Dinesh Baniya Kshatri

20

Sample 8086 Program – [6]

- Find the factorial of a number (N) which is entered via the keyboard. The result is to be saved in a memory location called FACT. Assume (N) is less than nine.

.MODEL SMALL	MOV DS, AX		MOV AX, 0001H	JNE BACK
.STACK 64	MOV AH, 01H		CMP BX, 0000H	FINAL: MOV FACT, AX
.DATA	INT 21H		JZ FINAL	MOV AH, 4CH
FACT DW 0000H	SUB AL, 30H	BACK:	MUL BX	INT 21H
.CODE	MOV AH, 00H		DEC BX	END
MOV AX, @DATA	MOV BX, AX		CMP BX, 0000H	

Er. Dinesh Baniya Kshatri

21

Sample 8086 Program – [7]

- The program finds the average of two bytes stored in memory locations labeled FIRST and SECOND. The average is to be stored in memory location AVG.

.MODEL SMALL	.CODE	MOV BH, 00H	INT 21H
.STACK 64	MOV AX, @DATA	ADD AX, BX	END
.DATA	MOV DS, AX	MOV CL, 02H	
FIRST DB 89H	MOV AL, FIRST	DIV CL	
SECOND DB 0CAH	MOV AH, 00H	MOV AVG, AL	
AVG DB ?	MOV BL, SECOND	MOV AH, 4CH	

Er. Dinesh Baniya Kshatri

22

Sample 8086 Program – [8]

- Count the number of ones in a byte stored in memory and save it in a memory location called COUNT.

.MODEL SMALL	MOV AX, @DATA		JNC NEXT	INT 21H
.STACK 64	MOV DS, AX		INC BL	END
.DATA	SUB BL, BL	NEXT:	DEC DL	
NUM DB 97H	MOV DL, 08H		JNZ AGAIN	
COUNT DB ?	MOV AL, NUM		MOV COUNT, BL	
.CODE	AGAIN: ROL AL, 1		MOV AH, 4CH	

Er. Dinesh Baniya Kshatri

23