



Experiment No. 08

Title: Write basic 8086 programs



Batch: B-1

Roll No.: 16010422234

Name: Chandana Ramesh Galgali

Experiment No.: 08**Aim: Write the following 8086 assembly programs:-**

- a) Print "Hello World"
- b) Convert an 8 bit binary number to decimal and display it

Resources needed: DOSBOX, TASM**Theory:**

The Intel 8086 is an 16 bit microprocessor designed in the mid-80s. It has a CISC instruction set. i.e. instructions of different categories take different numbers of clock cycles for execution. The assembly language programs shown here are an example of how assembly code looks like. Some of the statements are assembler directives, similar to compiler preprocessor directives.

DOSBOX is a free and open source DOS emulator. The DOS operating system was soon replaced by the Windows OS . DOS is no longer used or supported, but for instructional purposes it is still needed in academic institutions. Therefore the DOS emulator DOSBOX is used to run all legacy DOS programs, games etc. We will use the old TASM (Turbo Assembler) developed by Borland.

Program1: Print "Hello World"

```

; -----
; Program: Display the message "Hello World"
; Filename: Prog1.asm
; -----

.model tiny                ; com program
.stack 100h
.data
message db "Hello World$",'$'      ; Message to be displayed terminating with a $
.code                          ; code segment
org 100h                      ; code starts at offset 100h
start:
    mov ax, @data
    mov ds, ax

    call method1
    call crlf
    call method2
    mov ah,4ch               ; function to terminate
    int 21h                 ; Dos Interrupt

```

```

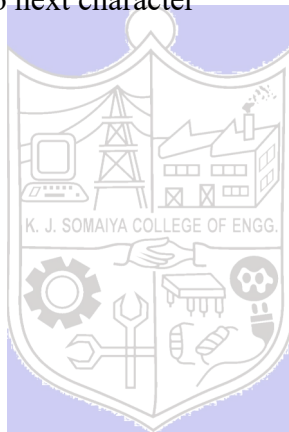
method1 proc near
    mov ah,09h                ; function to display a string
    lea dx, message           ; offset of Message string terminating with $
    int 21h                   ; dos interrupt
    ret
method1 endp

method2 proc near
    lea si, message           ; Make Si point to string address
loop1:
    mov dl, ds:[si]           ; Read First Character
    cmp dl,'$'
    je over                   ; if not '$' jump to loop1
    mov ah, 02h
    int 21h
    inc si                     ; Point to next character
    jmp loop1
over:
    ret
method2 endp

crlf proc near
    mov dl, 0ah
    mov ah, 02h
    int 21h
    mov dl, 0dh
    mov ah, 02h
    int 21h
    ret
crlf endp

end start

```



Program2: Convert an 8 bit binary number to decimal and display it.

```

; -----
; Program: Convert 8 bit binary number to decimal and display the result
; FileName: Prog2.asm
; -----

.model tiny                ; com program

```

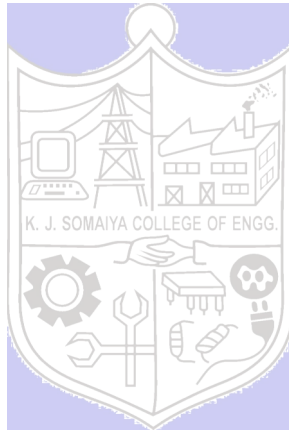
```

.stack 100h
.data
message db "Hello World$",'$'           ; Message to be displayed terminating with a $
bin8    db 01100100b
ans      db 0,0,0,'$'
hund     db 100
ten      db 10

.code                                     ; code segment
org 100h                                ; code starts at offset 100h
start:
    mov ax, @data
    mov ds, ax
convert:
    mov ah, 00
    mov al, bin8
    div hund
    lea si, ans
    mov ds:[si], al
    mov al, ah
    mov ah, 0
    div ten
    inc si
    mov ds:[si], al
    inc si
    mov ds:[si], ah
ascii:
    lea si, ans
    mov al, 30h
    add ds:[si], al
    add ds:[si+1], al
    add ds:[si+2], al

disp_ans:
    mov ah, 09h                        ; function to display a string
    lea dx, ans                        ; offset of Message string terminating with $
    int 21h                            ; dos interrupt
over:
    mov ah, 4ch                        ; function to terminate
    int 21h                            ; Dos Interrupt
end start

```



Procedure:

1. Download and install DOSBox and TASM in C drive
2. Type the programs given in this file and save them as .asm files.
3. Run DosBox
4. In the DosBox window, compile and run the program:
 - > tasm /l prog1.asm
 - > tlink prog1.obj
 - > prog1.exe
5. Verify the desired output

Observations and Results:

Both the programs should display the desired outputs in the DOSBOX window.

Paste the snapshots here.

In-Lab Program (For Practice)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

Address	Disassembly	Registers	Flags
cs:0000 B8AE44	mov ax,44AE	ax 4CAE	c=0
cs:0003 8ED8	mov ds,ax	bx F4D6	z=0
cs:0005 B409	mov ah,09	cx 0192	s=0
cs:0007 BA0000	mov dx,0000	dx 0000	o=0
cs:000A CD21	int 21	si F4D6	p=0
cs:000C B44C	mov ah,4C	di F5D8	a=0
cs:000E CD21	int 21	bp 0100	i=1
cs:0010 68656C	push 6C65	sp 0100	d=0
cs:0013 6C	insb	ds 44AE	
cs:0014 6F	outsw	es 449D	
cs:0015 20776F	and [bx+6F],dh	ss 44AF	
cs:0018 726C	jb 0086	cs 44AD	
cs:001A 642400	and fs:al,00	ip 000E	
cs:001D 0000	add [bx+si],al		
cs:001F 0000	add [bx+si],al		

es:0000 CD 20 FF 9F 00 EA FF FF = f 0
 es:0008 AD DE E5 01 00 15 AF 01 i 0 0 0 0
 es:0010 00 15 7D 02 1C 0F 92 01 0 0 0 0
 es:0018 01 01 01 00 02 FF FF FF 0 0 0 0
 es:0020 FF FF FF FF FF FF FF FF

ss:0108 0000
 ss:0106 0000
 ss:0104 0000
 ss:0102 0000
 ss:0100 0000

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Program1: Print "Hello World"

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[]-CPU 80486

Address	Instruction	Register/Value
cs:010E B44C	mov	ah,4C
cs:0110 CD21	int	21
cs:0112 B409	mov	ah,09
cs:0114 BA3A01	mov	dx,013A
cs:0117 CD21	int	21
cs:0119 C3	ret	
cs:011A BE3A01	mov	si,013A
cs:011D 8A14	mov	dl,[si]
cs:011F 80FA24	cmp	dl,24
cs:0122 7407	je	012B
cs:0124 B402	mov	ah,02
cs:0126 CD21	int	21
cs:0128 46	inc	si
cs:0129 EBF2	jmp	011D
cs:012B C3	ret	

es:0000 CD 20 FF 9F 00 EA FF FF = f 0
 es:0008 AD DE E5 01 00 15 AF 01 : 0 0 0 0 0 0 0 0
 es:0010 00 15 7D 02 1C 0F 92 01 : 0 0 0 0 0 0 0 0
 es:0018 01 01 01 00 02 FF FF FF : 0 0 0 0 0 0 0 0
 es:0020 FF FF FF FF FF FF FF FF : 0 0 0 0 0 0 0 0

ss:0108 0000
 ss:0106 0000
 ss:0104 0000
 ss:0102 0000
 ss:0100 0000

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Program2: Convert an 8 bit binary number to decimal and display it.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[]-CPU 80486

Address	Instruction	Register/Value
cs:0124 B030	mov	al,30
cs:0126 0004	add	[si],al
cs:0128 004401	add	[si+01],al
cs:012B 004402	add	[si+02],al
cs:012E B409	mov	ah,09
cs:0130 BA4801	mov	dx,0148
cs:0133 CD21	int	21
cs:0135 B44C	mov	ah,4C
cs:0137 CD21	int	21
cs:0139 004865	add	[bx+si+65],cl
cs:013C 6C	insb	
cs:013D 6C	insb	
cs:013E 6F	outsw	
cs:013F 20576F	and	[bx+6F],dl
cs:0142 726C	jb	01B0

es:0000 CD 20 FF 9F 00 EA FF FF = f 0
 es:0008 AD DE E5 01 00 15 AF 01 : 0 0 0 0 0 0 0 0
 es:0010 00 15 7D 02 1C 0F 92 01 : 0 0 0 0 0 0 0 0
 es:0018 01 01 01 00 02 FF FF FF : 0 0 0 0 0 0 0 0
 es:0020 FF FF FF FF FF FF FF FF : 0 0 0 0 0 0 0 0

ss:0108 0000
 ss:0106 0000
 ss:0104 0000
 ss:0102 0000
 ss:0100 0000

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Outcomes: Understand the fundamental concepts of microprocessors.

Conclusion:

Both programs were successfully written and executed using 8086 assembly language, achieving the desired objectives of the experiment.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. “Microprocessors and Interfacing”, by Douglas Hall Tata McGraw Hill 3rd edition.

