COMSATS University Islamabad, Lahore Campus

Department of Computer Engineering

Microprocessor Systems and Interacting (CPE342)

Course Instructor: Engr. Usman Rafique

Assignment __ONE__ Section: _FA22-BCE-B_

Submitted by: Abdullah Laeeq

Reg. number: FA22-BCE-026

Submitted on: ________________________________

Marks Obtained	Q1	Q2	Q3	Total

COMSATS UNIVERSITY ISLAMABAD, Lahore Campus

Department of Computer Engineering

Subject: Microprocessor Systems and Interfacing (CPE342)	**Batch: FA22-BCE-B**
Assignment No. ONE	**Total Marks: 30**
Handed over on: 28th Feb. 2025	**Submission Date: 7th March 2025 (In class)**
Student's Name: Abdullah Laeeq	
Registration Number: FA22-BCE-026	

Instructions:

- Provide your solution in the space provided against each problem
- Back side of each leaf is for rough work only
- Submission after the deadline will not be graded
- Do not use lead pencil in your solution

Note: The CPU referred to in this problem sheet is Intel 8086-88.

Problem 1 **10 Marks**

Write an assembly language program that computes the sum of four 16-bit numbers residing in SI, DI, AX, and DX. The sum must be stored in SS.

Solution:

```
mov si, 1234h
mov di, 5678h
mov ax, 9ABCh
mov dx, 0F012h

add ax, si
add ax, di
add ax, dx

mov ss, ax

mov bp, sp
sub bp, 2
mov [ss:bp], ax ;
```

Problem 2 **10 Marks**

Write an assembly language program that computes the average of all the 16-bit numbers found in all the segment registers. The result must be stored in SP. Use the minimum number of instructions.

Solution:

```
mov ax, cs
mov bx, ds
add ax, bx

mov bx, es
add ax, bx

mov bx, ss
add ax, bx

mov cl, 2
shr ax, cl ; AX = AX / 4 (logical shift right)

mov sp, ax
```

Problem 3 **10 Marks**

Write an assembly language program that complements the bits with bit number 0, 1, 3, 5, 11 and 14 of the words stored at 52A8H:4122H, AB44H:C23FH and 7CD0H:B234H. Make use of a subroutine named “BIT_MASK”.

Solution:

```
.data
; Bit mask for bits 0, 1, 3, 5, 11, 14
MASK dw 0100010000101011b
.code
; Set DS to point to data segment (to read MASK once)
mov ax, @data
mov ds, ax
; Save MASK in a register (since DS will change later)
mov cx, MASK   ; CX now holds the bit mask
; Complement bits in first word: 52A8:4122
mov ax, 052A8h
mov es, ax     ; Set ES to target segment
mov di, 04122h ; Offset within segment
call BIT_MASK
; Complement bits in second word: AB44:C23F
mov ax, 0AB44h
mov es, ax     ; Set ES to target segment
mov di, 0C23Fh ; Offset within segment
call BIT_MASK
; Complement bits in third word: 7CD0:B234
mov ax, 07CD0h
mov es, ax     ; Set ES to target segment
mov di, 0B234h ; Offset within segment
call BIT_MASK

; ======== Subroutine to complement bits ========
BIT_MASK proc
   ; Load the word from memory (ES:DI)
   mov bx, es:[di]

   ; Complement only the selected bits using CX (which holds MASK)
   xor bx, cx

   ; Store the modified word back to memory
   mov es:[di], bx

   ; Return to caller
   ret
BIT_MASK endp
```