

COE 205 Computer Organization & Assembly Language

Assignment 2: Registers, Flags, Data Allocation, and Basic I/O

Professor: Muhamed Mudawar

Q1. (5 pts) Show the content of the memory allocated in the data segment in hexadecimal for the following directives, and compute the offsets of *I*, *J*, and *L*.

```
.DATA
I   DB   1, 255
     DW   0FAh
     DD   -256
J   DB   '24'
K   EQU  24
L   DW   ?
     DW   OFFSET J
     DB   2 DUP ('*', 3 DUP('!'), '*')
```

A1. Address	Value	Comment
DS:0000	01	OFFSET I is assumed to be 0
0001	FF	
0002	FA	} DW 0FAh
0003	00	
0004	00	} DD -256
0005	FF	
0006	FF	
0007	FF	
0008	32	'2', OFFSET J = 8
0009	34	'4'
000A	?	OFFSET L = 0Ah = 10 in decimal
000B	?	OFTEN INITIALIZED TO 0
000C	08	
000D	00	
000E	28	\('
000F	2A	*'
0010	21	\!'
0011	21	\!'
0012	21	\!'
0013	2A	*'
0014	29	\)'
0015	28	
0016	2A	
0017	21	
0018	21	
0019	21	
001A	2A	
001B	29	

Q2. (5 pts) Suppose that you have the following initial register content:

AX=F2E9H BX=0002H CX=08A0H DX=F1E0H

(i) Show the contents of AX and the flags (*O, S, Z, A, P,* and *C*) at the end of executing the *ADD* instruction:

ADD AX, BX

Content of AX = F2EBh, O=0 (NO OVERFLOW), S=1 (NEGATIVE), Z=0 (NOT ZERO), A=0 (NO AUXILIARY CARRY), P=1 (EVEN PARITY IN LEAST SIGNIFICANT BYTE AL), C=0 (NO CARRY)

(ii) Show the contents of CX and the flags (*O,S,Z,A,P,* and *C*) at the end of executing the *SUB* instruction:

SUB CX, DX

Content of CX = 16C0h, O=0 (NO OVERFLOW), S=0 (POSITIVE), Z=0 (NOT ZERO), A=0 (NO AUXILIARY CARRY), P=1 (EVEN PARITY IN LEAST SIGNIFICANT BYTE CL), C=1 (CARRY)

Q3. (5 pts) Write an 8086 assembly program to (a) display a '?', (b) read two decimal digits, and (c) display them and their sum on the next line, with an appropriate message.

Sample Execution:

?27

2+7=9

A3. .model small

.stack 100H

.data

 endl DB 13, 10, '\$'

.code

.startup

 mov dl, '?' ; display '?'

 mov ah, 2

 int 21h

 mov ah, 1 ; read first digit

 int 21h

 mov bl,al ; save first digit in bl

 int 21h ; read second digit

 mov bh,al ; save second digit in bh

 add al,bh ; take the sum of digit codes

 sub al,30h ; convert it back to a digit code

 mov cl,al ; save sum in cl

 mov ah,9

 mov dx, offset endl ; display end of line

 int 21h

 mov ah, 2

 mov dl, bl ; display first digit

 int 21h

 mov dl, '+' ; display '+'

 int 21h

 mov dl,bh ; display second digit

 int 21h

