18.02.2019

Subject I

Explain what are and which is the role of the flags in the functioning of the 80x86 microprocessor.

Present the flags, their classification and exemplify the influence of the most important flags giving adequate sequences of instructions written in assembly language (at least 4 commented and explained examples – every example referring to a different flag).

Which are the instructions for direct accessing the flags?

Which are the instructions most involved/having the strong connection/that take into consideration the flag values?

How can we modify the configuration of the EFLAGS register if needed?

Which are the flags reacting to an overflow situation, which is their exact role and which are the rules for being set?

Why is there more than flag dealing with an overflow situation?

Subject II

a)

x dw -256, 256h

**x: 00 FF | 56 02**

256 = 0001 0000 0000b

-256 = 1111 0000 0000b = F00h

y dw 256|-256, 256h&256

**y: 00 FF | 00 00**

256 | -256 =1111 1111 0000 0000b = FF00h

256 = 0001 0000 0000b

256h = 0010 0101 0110b

256h&256 = 0000 0000 0000b = 0h

z db $-z, y-x

db ‘y’-‘x’, ‘y-x’

**z: 00 | 04 | 01 | ‘y’ ‘-‘ ‘x’**

$-z = 0

y-x = 4

‘y’-‘x’ = 1

a db 512>>2, -512<<2

**a: 80 | 00**

512 = 0010 0000 0000b

-512 = 1110 0000 0000b

512>>2 = 1000 0000b = 80h

-512<<2 = 1000 0000 0000b = 800h

b dw z-a, !(z-a)

**b: FA FF | 00 00**

z-a = -6? = 1111 … 1010b = F … FAh

!(z-a) = 0

c dd ($-b) + (d-$), $-2\*y+3

**syntax error – d is not declared yet**

d db -128, 128^(~128)

**d: 80 | FF**

128 = (0000) 1000 0000b

-128 = (1111) 1000 0000b = 80h

128^(~128) = 1000 0000b ^ 0111 1111b = 1111 1111b = FFh

e times 2 resw 6

times 2 dd 1234h, 5678h

**e: 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 – 12 words**

**34 12 00 00 | 78 56 00 00 | 34 12 00 00 | 78 56 00 00**

b)

mov bh, 7fh

cmp bh, al ; does nothing?

rcr ah, 1 ; ah stays the same after the last two instructions

sar ah, 7

**mov bh, 7fh**

Subject III

a)

mov eax, 200 EAX = 000000C8h AX = 00C8h = 200

mov ebx, 256h EBX = 00000256h BL = 56h = 86

idiv bl AH = 1C (%) AL = 02h (/)

**OF = CF = 0**

b)

mov ax, 256h AX = 0256h AH = 02h

mov dx, -1 DX = FFFFh DH = FFh

add ah, dh AH = AH + DH = 01h

**CF = 1, OF = 0**

c)

mov ax, ~(16h|32) AX = FFC9h AL = C9h

mov bx, 2000h>>4 BX = 0200h BH = 02h

imul bh AX = AL\*BH = 201\*2 = 402 = 192h => AX = FF92h ???

**CF = OF = 0**

~(16h|32) = ~(0001 0110 | 0010 0000) = ~(0011 0110) = (1111) 1100 1001 = FFC9h

2000h>>4 = 0010 0000 0000 0000>>4 = 0010 0000 0000 = 200h

d)

mov ax, 21<<7 AX = 0A80h AL = 80h

mov bh, 10h^3 BH = 13h

sub bh, al BH = BH + AL = 13h + 80h = 9\*16 + 3 = 147 = 93h

**CF = OF = 1**

21<<7 = 0001 0101 <<7 = 1010 1000 0000 = A80

10h^3 = 0001 0000 ^ 0000 0011 = 0001 0011 = 13h

e) !!!!!!!!!!!!????????????

shl bh, 8 BH = 0 ? maybe

add bx, ax BX = BX + AX

mov eax, ebx