

Name: Umeemah Hussain

Roll Number: 21L-1858

Class: BSCS-3A

## Assignment-1

Q.1

a.) add cx, bx

cx = 0x334A  $\Rightarrow$  3 3 4 A

bx = 0x45F1  $\Rightarrow$  4 5 F 1

0 7 9 3 B

ZF=0 ; SF=0

OF=0 ; CF=0

b.) add cx, bx

cx = 0x8934  $\Rightarrow$  8 9 3 4

bx = 0x45F1  $\Rightarrow$  4 5 F 1

0 0 F 2 5

SF=0 ; ZF=0

OF=0 ; CF=0

c.) sub bx, 6

bx = 0x45F1  $\Rightarrow$  4 5 F 1  
6

CF=0 ; ZF=0

4 5 E B

SF=0 ; OF=0

Q.2

a.) 0xb900

b.) 0x4567

c.) 0xAA99

upper b9

upper 45

upper AA

lower 00

lower 67

lower 99

Q.3

a.) FFFF : 4312

b.) 1DEF : 0001

F F F F 0

1 D E F 0

4 3 1 2

0 0 0 1

1 0 4 3 0 2

0 1 D E F 1

c.) 14FF : 1111

1 4 F F 0

1 1 1 1

0 1 6 1 0 1



Date:    /    /   

Q.4

[org 0x100]

mov ax, 12

mov cx, 6

mov bx, 20

l1:

add cx, ax

sub bx, 1

jne l1

mov ax, cx

mov ax, 0x400

int 0x21

Q.5

a.)  $bp - di$

- This is NOT valid
- Only addition of base registers with index registers is allowed

b.)  $bp + si$

$$\begin{array}{lcl} bp = 0x0220 & \Rightarrow & 0x0220 \\ si = 0x0110 & & 0x0110 \\ & & \hline & & 0x0330 \end{array}$$

c.)  $bx - 0x12$

$$\begin{array}{l} 0x0034 \\ 0x0012 \\ \hline 0x0022 \end{array}$$

d.)  $bx + bp$

- This is NOT valid
- We can not have two base registers in memory access

e.)  $bx + ip$

- This is NOT valid
- No memory access can be performed through IP.



f.)  $bx + di$

0x0034

0x1101

0x1135

Q. 6

a.)  $[cs:bx + si]$

Whole memory / Physical address wraparound

0x479b9

b.)  $[bp + di + 10]$

Segment Wraparound

4c177

Q. 7

a.)  $\text{mov ip, bx}$

IP can't be overwritten

$\text{mov ax, bx}$

b.)  $\text{mov byte bx, [ip]}$

IP can't be manually accessed

$\text{mov bx, cx}$

c.) `mov si, al`

size mismatch

`mov bl, al`

d.) `mov ax, [bx+bp+100]`

- `bx` and `bp` can not be added as they are both base registers

- `mov ax, [bx+si+100]`

Q.8

`OF = 0 ; CF = 0 ; PF = 0 ; SF = 1`

Q.9

There is no logical error in the code

`org [0x100]`

`jmp start`

`num1: db -2, -4, -5, 1, 3`

`start:`

`mov al, [num1]`

`mov bl, [num1+1]`

`add al, bl`

`mov bl, [num1+2]`

`add al, bl`

`mov bl, [num1+3]`

`add al, bl`

`cmp byte [num1+4], 4`

`jg end`

`mov byte [num1+4], 2`

`end: mov ax, 0x4000`  
`int 0x21`



Q. 10

[org 0x100]

mov ax, num1

mov bx, [num1+1]

mov cx, ax

mov ax, bx

mov bx, cx

mov [num1], ax

mov [num1+1], bx

mov cx, [num1+2]

mov bx, [num1+3]

mov cx, ax

mov ax, bx

mov bx, cx

mov [num1], ax

mov [num1+1], bx

mov ax, 0x4c00

int 0x21

num1: db 1, 2, 3, 4

Q. 11

```
[org 0x0100]
```

```
mov ax, [array 1]
```

```
mov [min], ax
```

start:

```
mov ax, [array 1+2]
```

```
mov bx, [min]
```

```
cmp ax, bx
```

```
jge end
```



Date:    /    /   

```
mov ax, [array1+6]
```

```
mov bx, [min]
```

```
cmp ax, bx
```

```
jge end
```

```
mov ax, [array1+8]
```

```
mov bx, [min]
```

```
cmp ax, bx
```

```
jge end
```

```
end: mov [min], ax
```

```
mov bx, [min]
```

```
mov ax, 0x4c00
```

```
int 0x21
```

```
array1: dw 5, 3, -8, 2, 5
```

```
min: dw 0
```