

# Tutorial 4

## Answers

CO 2103 Assembly Language

# Program Control - 1

- **Task 1:** Given that  $AX=1122h$ ,  $BX=ddddh$ ,  $CX=eeffh$  initially, determine for each of the following program segments (independently), whether **L1** or **L2** will be executed:

i. `xor ax,ax`  
`jz L2`  
`L1: ;instructions`  
`jmp L3`  
`L2: ;instructions`  
`L3: ;ending`

**L2**

ii. `cmp ax,bx`  
`jge L2`  
`L1: ;instructions`  
`jmp L3`  
`L2: ;instructions`  
`L3: ;ending`

**L1**

iii. `add ax,bx`  
`cmp ax,cx`  
`je L2`  
`L1: ;instructions`  
`jmp L3`  
`L2: ;instructions`  
`L3: ;ending`

**L2**

iv. `sub ax,bx`  
`jns L2`  
`L1: ;instructions`  
`jmp L3`  
`L2: ;instructions`  
`L3: ;ending`

**L1**

# Program Control - 2

- Task 2:** Given that  $AX=3$ ,  $BX=7$ ,  $CX=8$  initially, determine how many times will **L1** be executed in the following program segments (independently):

i.

```
L1: dec ax
    jnz L1
    mov ax, 3
    dec bx
    jnz L1
    ;ending
```

$3 \times 7 = 21$

ii.

```
L1: dec bx
    cmp bx, ax
    jne L1
    ;ending
```

$7 - 3 = 4$

iii.

```
L1: dec ax
    loop L1
    ;ending
```

$CX = 8$

iv.

```
L1: dec ax
    loopnz L1
    ;ending
```

$AX < CX, 3$

v.

```
L1: add ax, 1
    cmp ax, bx
    loopne L1
    ;ending
```

$(7 - 3) = 4 < CX, 4$

# Simple AL Programming - 1

- **Task 3:**

```
;Task 3.1: a = x * (y + z)
xor dx,dx          ;dx=0
xor ax,ax          ;ax=0
add ax,y           ;ax=y
add ax,z           ;ax=y+z
mov bx,x           ;bx=(x)
mul bx             ;dx ax=ax * bx=x * (y+z)
;save result
mov si,offset a
mov [si],ax        ;lower word of the result
mov [si+2],dx      ;upper word of the result
;dd ptr (a) = dx ax
```

```
;Task 3.2: a = x^4
mov cx,4           ;power
xor dx,dx          ;dx=0
xor ax,ax          ;ax=0
mov bx,x           ;bx=x, i.e. base
L1: mul bx         ;dx ax=ax*bx
loop L1            ;multiply cx times
;save result
mov si,offset a
mov [si],ax        ;lower word of the result
mov [si+2],dx      ;upper word of the result
;dd ptr (a) = dx ax
```

# Simple AL Programming - 2

- **Task 3 con't:**

```
;Task 3.3: a = w - (x * y) + z
xor dx,dx          ;dx=0
xor ax,ax          ;ax=0
mov ax,x           ;ax=x
mov bx,y           ;bx=y
mul bx             ;dx ax=ax*bx=x*y
;for simplicity, assume result fits in 16-bit,
;i.e. ax while dx=0
mov bx,w           ;bx=w
sub bx,ax          ;bx=bx-ax=w-(x*y)
add bx,z           ;bx=bx+z=w-(x*y)+z
;what changes do you have to make if result exceeds 16-bit?
;save result
mov word ptr a,bx
```

- **Task 4:**

```
;Task 4.1: print the string "working on tutorial 3" on the screen
mov ah,9      ;function: print string
mov dx,offset msg
              ;DS:DX points to string head, i.e. msg
int 21h
```

```
.data
msg db "working on tutorial 3"
```

```
;Task 4.2: read a character from the keyboard
mov ah,01h   ;function: read character with echo
int 21h      ;al = ascii of character input
```