



Computer Organization and Assembly Language

Lab 6

Student Name	Umamah Hussain
Roll Number	21L-1858
Instructor	Hazoor Ahmad
Class	CS3
Section	A1
Semester	Fall 2022

Fast School of Computing

FAST-NU, Lahore, Pakistan

Activity 1:

Assembly Language Code

```
1  [org 0x100]
2  call clearscreen
3  mov ax, 0x4c00
4  int 21h
5  clearscreen: push es
6  push ax
7  push di
8  mov ax, 0xb800
9  mov es, ax
10 mov di, 0
11 nextchar: mov word [es:di], 0x0720
12 add di, 2
13 cmp di, 4000
14 jne nextchar
15 pop di
16 pop ax
17 pop es
18 ret
19
```

Debugging Screenshots



Activity 2

Assembly Language Code

```
[org 0x100]
call ascii_convert
mov dx, [n1]
mov dx, [n2]
mov dx, [n3]
mov dx, [n4]
mov ax, 0x4c00
int 21h
ascii_convert:
mov cx, 12
mov ax, 5306
mov dx, 0xf000
and ax, dx
l1:shr ax, 1
loop l1
add ax, 0x30
mov [n1], ax
mov cx, 8
mov ax, 5306
mov dx, 0x0f00
```

```
and ax, dx
12:shr ax, 1
loop 12
add ax, 0x30
mov [n2], ax
mov cx, 4
mov ax, 5306
mov dx, 0x00f0
and ax, dx
13:shr ax, 1
loop 13
add ax, 0x30
mov [n3], ax
mov ax, 5306
mov dx, 0x000f
and ax, dx
add ax, 0x30
mov [n4], ax
end:ret
temp: db 10, 100, 1000, 10000
n1: dw 0
n2: dw 0
```

```
n3: dw 0
n4: dw 0
```

```
n4: dw 0
```

Debugging Screenshots

```

DOS D:\DOS\U.74, Cpu speed: 3000 cycles, Frameskip U, Program: AFD
AX 003A SI 0000 CS 19F5 IP 010F Stack +0 0000 Flags 7204
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 003B SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 1 0

CMD >

010B 8B166E01 MOV DX,[016E]
010F 8B167001 MOV DX,[0170]
0113 B8004C MOV AX,4C00
0116 CD21 INT 21
0118 B90C00 MOV CX,000C
011B B8BA14 MOV AX,14BA
011E BA00F0 MOV DX,F000
0121 21D0 AND AX,DX
0123 D1EB SHR AX,1

01 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 00 = f.n ; |.+.
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 00 .....f. ....
DS:0020 01 00 01 00 01 00 01 00 01 FF FF FF EB 19 E6 11 ..... δ.μ.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 6.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Activity 3:

Assembly Language Code

```
[org 0x100]
call clearscreen
call ascii_convert
```

```
mov ax, string
push ax
push word [length]
call printstr
mov ax, 0x4c00
int 21h

clearscreen: push es
push ax
push di
mov ax, 0xb800
mov es, ax
mov di, 0
nextchar: mov word [es:di], 0x0720
add di, 2
cmp di, 4000
jne nextchar
pop di
pop ax
pop es
ret

ascii_convert:
mov cx, 12
```

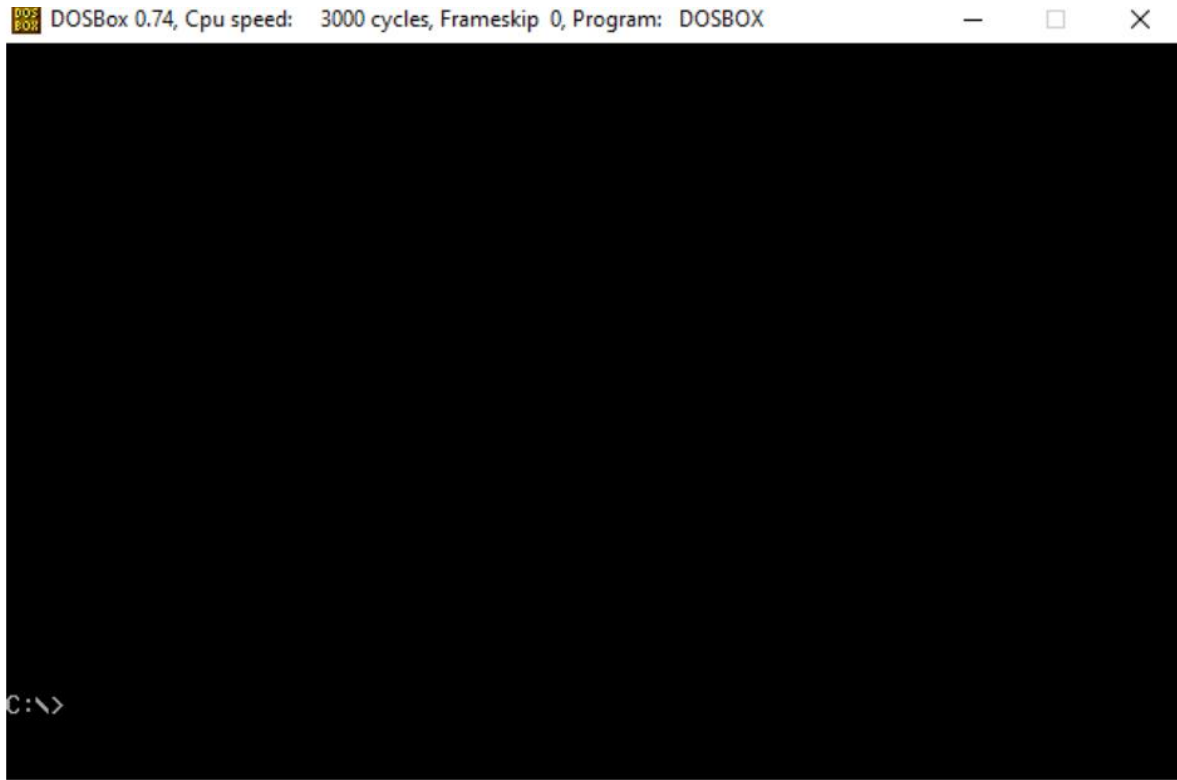
```
mov ax, 5306
mov dx, 0xf000
and ax, dx
l1:shr ax, 1
loop l1
add ax, 0x30
mov [n1], ax
mov cx, 8
mov ax, 5306
mov dx, 0x0f00
and ax, dx
l2:shr ax, 1
loop l2
add ax, 0x30
mov [n2], ax
mov cx, 4
mov ax, 5306
mov dx, 0x00f0
and ax, dx
l3:shr ax, 1
loop l3
add ax, 0x30
```



```
mov [n3], ax
mov ax, 5306
mov dx, 0x000f
and ax, dx
add ax, 0x30
mov [n4], ax
end:ret
printstr: push bp
mov bp, sp
push es
push ax
push cx
push si
push di
mov ax, 0xb800
mov es, ax
mov di, 0
mov si, [bp+6]
mov cx, [bp+4]
mov ah, 0x07
next: mov al, [si]
mov [es:di], ax
```

```
add di, 2
add si, 1
loop nex
pop di
pop si
pop cx
pop ax
pop es
pop bp
ret 4
string: db "Umamah Hussain - 211-1858"
length: db 22
n1: dw 0
n2: dw 0
n3: dw 0
n4: dw 0
```

Debugging Screenshots



Activity 4

Assembly Language Code

```
[org 0x0100]
mov ax, 0x5306
call seperate
call summation
mov cx, [r]
mov bx, [1]
mov ax, 0x4c00
```

```
int 21h
summation:
mov ax, 0
mov ax, [n1]
add ax, [n2]
add ax, [n3]
add ax, [n4]
shr ax, 2
mov[r],ax
add ax, 2
mov[l],ax
ret
seperate:
mov cx, 12
mov ax, 5306
mov dx, 0xf000
and ax, dx
l1:shr ax, 1
loop l1
mov [n1], ax
mov cx, 8
mov ax, 5306
```

```
mov dx, 0x0f00
and ax, dx
12:shr ax, 1
loop 12
mov [n2], ax
mov cx, 4
mov ax, 5306
mov dx, 0x00f0
and ax, dx
13:shr ax, 1
loop 13
mov [n3], ax
mov ax, 5306
mov dx, 0x000f
and ax, dx
mov [n4], ax
end:ret
n1: dw 0
n2: dw 0
n3: dw 0
n4: dw 0
r: dw 0
```

```
1: dw 0
```

Debugging Screenshots

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 4C00 SI 0000 CS 19F5 IP 0114 Stack +0 0000 Flags 7200
 BX 0008 DI 0000 DS 19F5 +2 20CD
 CX 0006 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
 DX 000F SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

CMD >

0111 B8004C	MOV	AX, 4C00
0114 CD21	INT	21
0116 B80000	MOV	AX, 0000
0119 A17701	MOV	AX, [0177]
011C 03067901	ADD	AX, [0179]
0120 03067B01	ADD	AX, [017B]
0124 03067D01	ADD	AX, [017D]
0128 C1E802	SHR	AX, 02
012B A37F01	MOV	[017F], AX

	0	1	2	3	4	5	6	7
DS:0000	CD	20	FF	9F	00	EA	F0	FE
DS:0008	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01
DS:0018	01	01	01	00	FF	00	01	00
DS:0020	01	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19
DS:0038	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00
DS:0048	00	00	00	00	00	00	00	00

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	01	00	FF	00	01	00
DS:0020	01	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri

Activity 5

Assembly Language Code

```
[org 0x100]
mov ax, 0x5306
call seperate
call summation
```

```
mov ax, 0x4c00
```

```
int 21h
```

```
summation:
```

```
mov ax, 0
```

```
mov ax, [n1]
```

```
add ax, [n2]
```

```
add ax, [n3]
```

```
add ax, [n4]
```

```
shr ax, 2
```

```
mov[r],ax
```

```
add ax, 2
```

```
mov[l],ax
```

```
ret
```

```
seperate:
```

```
mov cx, 12
```

```
mov ax, 5306
```

```
mov dx, 0xf000
```

```
and ax, dx
```

```
l1:shr ax, 1
```

```
loop l1
```

```
mov [n1], ax
```

```
mov cx, 8
```

```
mov ax, 5306
mov dx, 0x0f00
and ax, dx
12:shr ax, 1
loop 12
mov [n2], ax
mov cx, 4
mov ax, 5306
mov dx, 0x00f0
and ax, dx
13:shr ax, 1
loop 13
mov [n3], ax
mov ax, 5306
mov dx, 0x000f
and ax, dx
mov [n4], ax
end:ret
addition:
mov ax, [f]
mov bx, word [result]
add bx, ax
```



```
mov [reres], bx
mov bx, word [result + 2]
mov ax, [f+2]
adc bx, ax
mov [reres + 2], bx
ret
n1: dw 0
n2: dw 0
n3: dw 0
n4: dw 0
r: dw 0
f: dd 0
```

Debugging Screenshots

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 4C00	SI 0000	CS 19F5	IP 0114	Stack +0 0000	Flags 7200
BX 0008	DI 0000	DS 19F5		+2 20CD	
CX 0006	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 000F	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 0 0

CMD >

0111 B8004C	MOV	AX,4C00	DS:0000	CD 20 FF 9F 00 EA F0 FE
0114 CD21	INT	21	DS:0008	AD DE 1B 05 C5 06 00 00
0116 B80000	MOV	AX,0000	DS:0010	18 01 10 01 18 01 92 01
0119 A17701	MOV	AX,[0177]	DS:0018	01 01 01 00 FF 00 01 00
011C 03067901	ADD	AX,[0179]	DS:0020	01 FF FF FF FF FF FF FF
0120 03067B01	ADD	AX,[017B]	DS:0028	FF FF FF FF EB 19 C0 11
0124 03067D01	ADD	AX,[017D]	DS:0030	A2 01 14 00 18 00 F5 19
0128 C1E802	SHR	AX,02	DS:0038	FF FF FF FF 00 00 00 00
012B A37F01	MOV	[017F],AX	DS:0040	05 00 00 00 00 00 00 00
			DS:0048	00 00 00 00 00 00 00 00

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00	= f.Ω≡■ ↓ ..†...
DS:0010	18	01	10	01	18	01	92	01	01	01	00	FF	00	01	00	ff.
DS:0020	01	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11		. δ.L.
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00	6.....J.
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri