

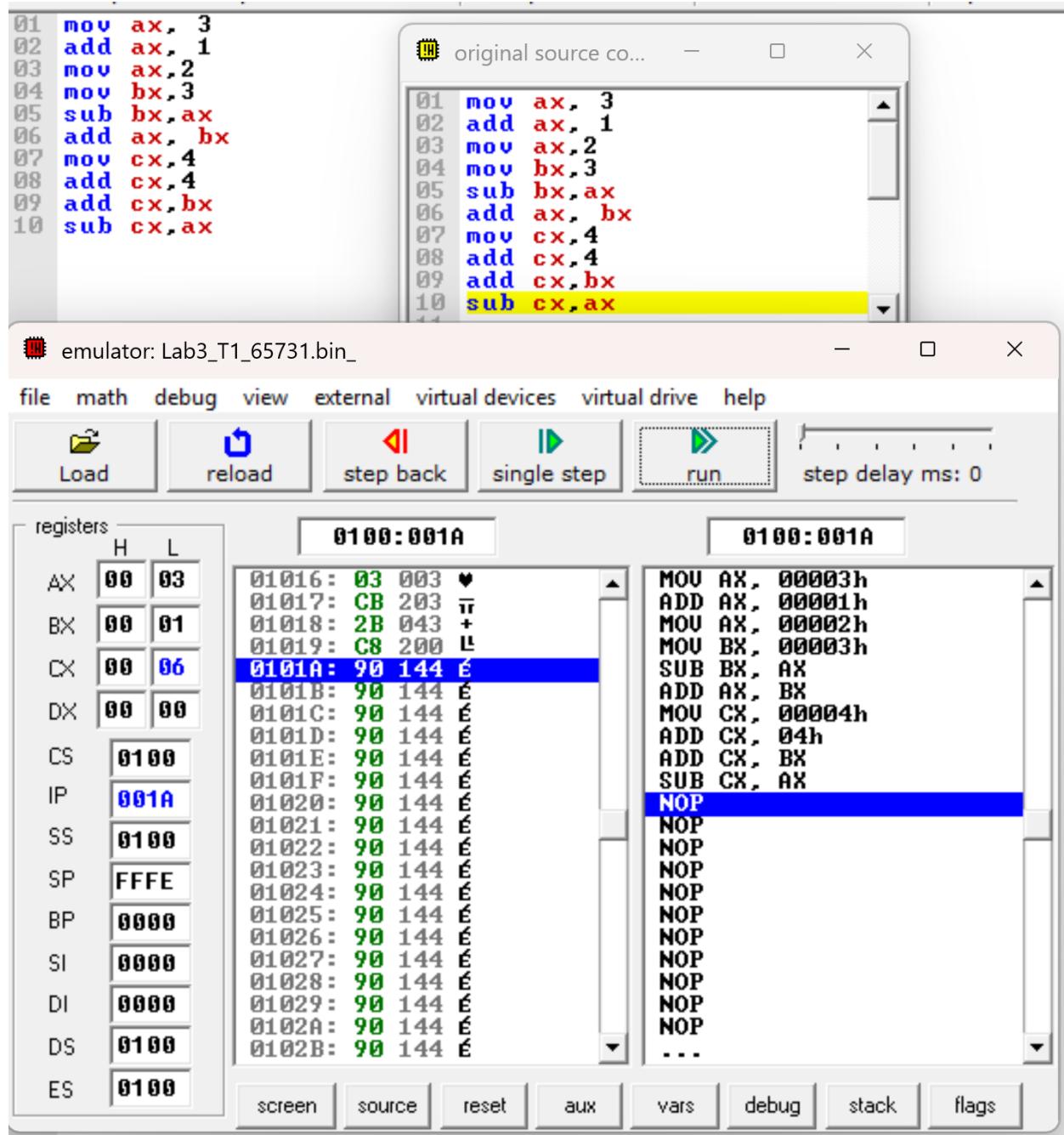
Riphah International University

Babar Ali

65731

3rd Semester

**Computer Organization & AL
Lab 03**



The screenshot shows a debugger interface with two windows and several toolbars.

Top Window: A code editor titled "original source co..." showing assembly code. The code consists of 10 lines, each starting with a byte address (e.g., 01, 02, ..., 10) followed by an instruction and operands. The code is identical to the one in the registers window.

```
01 mov ax,3  
02 add ax,4  
03 mov bx,5  
04 add bx,7  
05 mov bx,ax  
06 mov ax,bx  
07 sub ax,1  
08 sub ax,2  
09 sub bx,1  
10 sub bx,2
```

Main Window: Titled "emulator: Lab3_T2_65731.bin_".

Toolbar: Contains buttons for "Load", "reload", "step back", "single step", "run", and "step delay ms: 0".

Registers: A table showing CPU register values. The AX register is highlighted with a yellow background.

	H	L
AX	00	04
BX	00	04
CX	00	00
DX	00	00
CS	0100	
IP	001D	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

Memory Dump: Two panes showing memory dump at address 0100:001D. The left pane shows raw hex and ASCII data, and the right pane shows the assembly instructions.

	0100:001D	0100:001D
01016:	83 131 à	MOU AX, 00003h
01017:	EB 235 δ	ADD AX, 00004h
01018:	01 001 ⊕	MOU BX, 00005h
01019:	83 131 à	ADD BX, 07h
0101A:	EB 235 δ	MOU BX, AX
0101B:	02 002 ⊖	MOU AX, BX
0101C:	90 144 É	SUB AX, 00001h
0101D:	90 144 É	SUB AX, 00002h
0101E:	90 144 É	SUB BX, 01h
0101F:	90 144 É	SUB BX, 02h
01020:	90 144 É	NOP
01021:	90 144 É	NOP
01022:	90 144 É	NOP
01023:	90 144 É	NOP
01024:	90 144 É	NOP
01025:	90 144 É	NOP
01026:	90 144 É	NOP
01027:	90 144 É	NOP
01028:	90 144 É	NOP
01029:	90 144 É	NOP
0102A:	90 144 É	NOP
0102B:	90 144 É	NOP
		...

Bottom Buttons: screen, source, reset, aux, vars, debug, stack, flags.

The screenshot shows a debugger interface with several windows:

- Assembly View:** Shows assembly code from line 01 to 10. Lines 01 through 09 are highlighted in red, and line 10 is highlighted in yellow.
- Registers View:** Displays CPU registers:

	H	L
AX	FF	FF
BX	02	5C
CX	00	06
DX	00	00
CS	0100	
IP	001C	
SS	0100	
SP	FFFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	
- Memory Dump View:** Displays memory starting at address 0100:001C. The instruction at 0101C is highlighted in blue. The assembly code for the dump area is:

```
0100:001C: MOU AX, 01234h  
0100:001C: MOU BX, 00256h  
0100:001C: MOU CX, 00004h  
0100:001C: ADD CX, 02h  
0100:001C: MOU AX, 00008h  
0100:001C: ADD BX, CX  
0100:001C: ADD AX, 00001h  
0100:001C: SUB AX, 00004h  
0100:001C: SUB AX, CX  
0100:001C: MOU CX, 00006h  
0100:001C: NOP  
0100:001C: ...
```
- Control Buttons:** Includes buttons for Load, Reload, Step Back, Single Step, Run, and Step Delay ms: 0.
- Bottom Navigation:** Buttons for screen, source, reset, aux, vars, debug, stack, and flags.

The screenshot shows a debugger interface with two windows and several toolbars.

Top Window: A source code editor titled "original source co...". It displays assembly code with line numbers from 01 to 10. Lines 01 through 09 are in blue, and line 10 is in yellow, indicating it is the current instruction being executed.

```
01 mov ax,4  
02 mov bx,4  
03 mov cx,14  
04 mov dx,2  
05 add ax,cx  
06 add dx,bx  
07 mov cx,6  
08 sub cx,4  
09 mov cx,dx  
10 add cx,dx
```

Bottom Window: An emulator window titled "emulator: Lab3_T4_65731.bin_".

Toolbar: Includes "file", "math", "debug", "view", "external", "virtual devices", "virtual drive", and "help".

Buttons: "Load", "reload", "step back", "single step", "run", and "step delay ms: 0".

Registers: A table showing CPU register values:

	H	L
AX	00	12
BX	00	04
CX	00	0C
DX	00	06
CS	0100	
IP	001B	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

Memory Dump: Two panes showing memory dump at address 0100:001B. The left pane shows raw hex values, and the right pane shows the corresponding assembly instructions.

	0100:001B	0100:001B
01016:	8B 139 i	MOU AX, 00004h
01017:	CA 202 II	MOU BX, 00004h
01018:	03 003 ♥	MOU CX, 0000Eh
01019:	CA 202 II	MOU DX, 00002h
0101A:	90 144 E	ADD AX, CX
0101B:	90 144 E	ADD DX, BX
0101C:	90 144 E	MOU CX, 00006h
0101D:	90 144 E	SUB CX, 04h
0101E:	90 144 E	MOU CX, DX
0101F:	90 144 E	ADD CX, DX
01020:	90 144 E	NOP
01021:	90 144 E	NOP
01022:	90 144 E	NOP
01023:	90 144 E	NOP
01024:	90 144 E	NOP
01025:	90 144 E	NOP
01026:	90 144 E	NOP
01027:	90 144 E	NOP
01028:	90 144 E	NOP
01029:	90 144 E	NOP
0102A:	90 144 E	NOP
0102B:	90 144 E	NOP
		...

Tool Buttons: screen, source, reset, aux, vars, debug, stack, flags.

The screenshot shows a debugger interface with two windows. The top window displays assembly code with some instructions highlighted in yellow. The bottom window shows the register state and a memory dump.

Assembly Code (Top Window):

```
01 add ax,120h  
02 mov cx,240h  
03 mov dx,520h  
04 add dx,ax  
05 mov bx,340h  
06 add bx,dx  
07 mov ax,4  
08 mov cx,6  
09 sub cx,ax  
10 mov cx,8  
11 mov ax,3  
12 add ax,4  
13 mov bx,5  
14 add bx,7  
15 mov bx,ax  
16 mov ax,bx  
17 sub ax,1  
18 sub ax,2  
19 sub bx,1  
20 sub bx,2
```

Registers (Bottom Window):

	H	L
AX	00	04
BX	00	04
CX	00	08
DX	06	40
CS	0100	
IP	0038	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

Memory Dump (Bottom Window):

	0100:0038	0100:0038
0102E:	2D 045	-
0102F:	02 002	NULL
01030:	00 000	â
01031:	83 131	â
01032:	EB 235	δ
01033:	01 001	Θ
01034:	83 131	â
01035:	EB 235	δ
01036:	02 002	Θ
01037:	90 144	É
01038:	90 144	É
01039:	90 144	É
0103A:	90 144	É
0103B:	90 144	É
0103C:	90 144	É
0103D:	90 144	É
0103E:	90 144	É
0103F:	90 144	É
01040:	90 144	É
01041:	90 144	É
01042:	90 144	É
01043:	90 144	É

Control Buttons (Bottom Window):

- Load
- reload
- step back
- single step
- run
- step delay ms: 0

Other Buttons (Bottom Window):

- screen
- source
- reset
- aux
- vars
- debug
- stack
- flags

original source code

```
01 mov ax, 3
02 add ax, 1
03 mov ax, 2
04 mov bx, 3
05 sub bx, ax
06 add ax, bx
07 mov ax, 1136h
08 mov bx, 23h
09 add ax, bx
10 mov cx, 45h
11 sub ax, cx
```

emulator: noname.bin_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0000

Address	OpCode	OpName
01000:	B8 184	MOU AX, 00003h
01001:	03 003	ADD AX, 00001h
01002:	00 000	MOU AX, 00002h
01003:	05 005	SUB BX, AX
01004:	01 001	ADD AX, BX
01005:	00 000	MOU AX, 01136h
01006:	B8 184	MOU BX, 00023h
01007:	02 002	ADD AX, BX
01008:	00 000	MOU CX, 00045h
01009:	BB 187	SUB AX, CX
0100A:	03 003	NOP
0100B:	00 000	NOP
0100C:	2B 043	NOP
0100D:	D8 216	NOP
0100E:	03 003	NOP
0100F:	C3 195	NOP
01010:	B8 184	NOP
01011:	36 054	NOP
01012:	11 017	NOP
01013:	BB 187	NOP
01014:	23 035	...

0100:0000

screen source reset aux vars debug stack flags