



Repository: Assembly-Language-927



Name:
Junaid Ahmad



Assignment#1

Technical Example of

- A) Input
- B) Process
- C) Output

Technical Example of

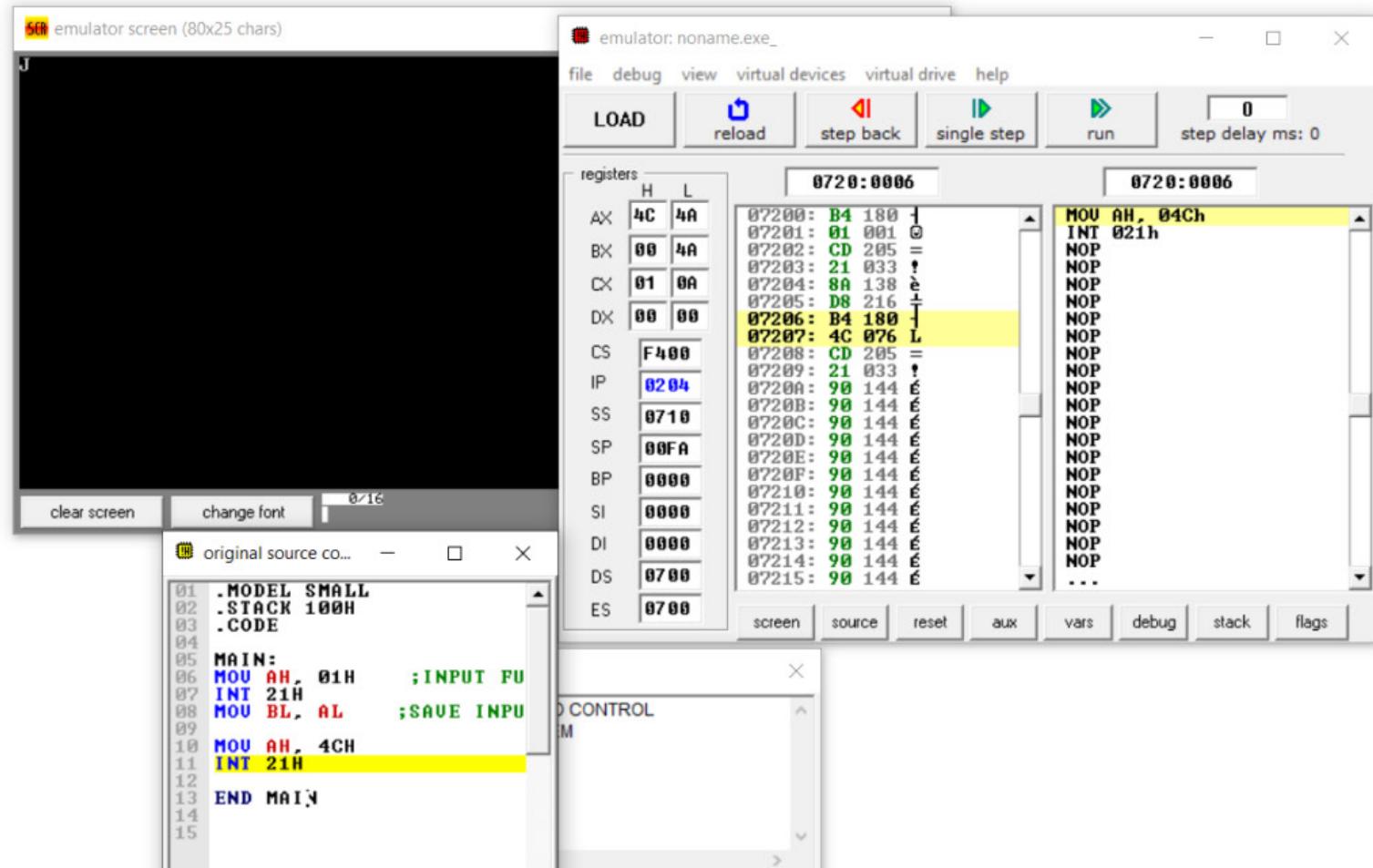
A) Input

Takes single character input from user

Stores input in BL register

Exits program after input

```
01 .MODEL SMALL
02 .STACK 100H
03 .CODE
04
05 MAIN:
06     MOU AH, 01H ;INPUT FUNCTION
07     INT 21H
08     MOU BL, AL ;SAVE INPUT IN BL
09
10    MOU AH, 4CH
11    INT 21H
12
13 END MAIN
```



Technical Example of

B) Process

Loads 05h and 03h into registers

Adds values and stores result in AL

Demonstrates basic arithmetic operation

The screenshot shows a debugger interface with the following components:

- Left Panel (Assembly View):** Displays the assembly code for the program. The code includes:
 - .MODEL SMALL
 - .STACK 100H
 - .CODE
 - MAIN:
 - MOU AL, 05H ;FIRST NUMBER
 - MOU BL, 03H ;SECOND NUMBER
 - ADD AL, BL ;AL = AL + BL = 8
 - MOU AH, 4CH
 - INT 21H
 - END MAIN
- Middle Panel (Registers View):** Shows the CPU registers with their current values:

	H	L
AX	4C	08
BX	00	03
CX	01	0A
DX	00	00
CS	0720	
IP	001E	
SS	0710	
SP	0100	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	
- Right Panel (Memory Dump):** Displays the memory dump starting at address 0720:001E. The memory contains the following bytes:

Address	Value	Content
07216:	90	NOP
07217:	90	NOP
07218:	90	NOP
07219:	90	NOP
0721A:	90	HLT
0721B:	90	HLT
0721C:	90	HLT
0721D:	90	HLT
0721E:	F4	INT 21H
0721F:	00	NULL
07220:	00	NULL
07221:	00	NULL
07222:	00	NULL
07223:	00	NULL
07224:	00	NULL
07225:	00	NULL
07226:	00	NULL
07227:	00	NULL
07228:	00	NULL
07229:	00	NULL
0722A:	00	NULL
0722B:	00	NULL

Technical Example of

C) Output

Displays "HELLO!" message on screen

Uses data segment for string storage

Implements string output function

The screenshot shows a debugger interface with several windows:

- Assembly Window:** Displays the assembly code for the program. The code initializes the stack, defines a message string 'MSG DB 'HELLO!\$'', and implements a main routine that prints the string using INT 21H and exits with INT 21H.
- Registers Window:** Shows the CPU registers:

	H	L
AX	4C	24
BX	00	00
CX	01	20
DX	00	00
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
- Stack Window:** Shows the stack contents starting at address F400:0204, which contains the string 'HELLO!' followed by null bytes.
- Memory Dump Window:** Shows the memory dump starting at address F400:0204, displaying the same data as the stack window.
- Registers View:** Shows the current state of the registers.
- Registers View:** Shows the current state of the registers.

Overview & Types of Register

Register Type	Registers	Bits	Code Example (EMU-8086)
General Purpose (Input Role)	AX, BX, CX, DX	16-bit	MOV AX, [var1] IN AL, 60h
Index — Source	SI	16-bit	MOV SI,OFFSET src LODSB ;AL<-[SI]
Segment — Data	DS	16-bit	MOV AX,@data MOV DS,AX
Accumulator (ALU Center)	AX / AL / AH	16/8-bit	ADD AX,BX MUL CX ;DX:AX=AX*CX
Count Register (Loop Control)	CX	16-bit	MOV CX,10 lbl: ADD AX,BX LOOP lbl
Instr. Pointer	IP	16-bit	JMP label CALL myProc RET
Segment — Extra	ES	16-bit	MOV AX,0B800h MOV ES,AX STOSW
Stack Pointer (Return Values)	SP	16-bit	PUSH param CALL func ADD SP,2 ;cleanup
DX — I/O Port & High Word	DX	16-bit	MUL BX ;DX:AX=res OUT DX,AL
SI + DI Together (Block Transfer)	SI, DI	16-bit	MOV CX,100 CLD REP MOVSB