Laboratory Report

Activity #1

<Write the calculations and solutions performed in Activity #1. Include sources codes and schematic diagrams if required in the e

Laboratory Exercise No.:	2	**Date Performed:**	09/18/2025
Laboratory Exercise Title:	Addressing Modes		
Name of Student(s):	Christian Jay Y. Gallardo, Jhon Fil Tizon	**Document Version:**	1

Activity #1:

Instruction	Register and memory value after the instruction is executed
MOV AL, 25	AX: 0019H, IP: 0102H
MOV AX, 2345	AX: 0929H, IP: 0105H
MOV BX, AX	BX: 0929H, IP: 0107H
MOV CL, AL	CX: 0029H, IP: 0109H
MOV AL, DATA1	AX: 0925H, 010CH
MOV AX, DATA2	AX: 1234H, IP: 010F
MOV DATA3, AL	IP: 0112H
MOV DATA4, AX	IP:0115H
MOV BX, OFFSET DATA5	BX: 0131H, IP: 0118H
MOV AX, [BX]	AX: 2345H, 011AH
MOV DI, 02H	IP: 011D, DI: 0002H
MOV AX, [BX+DI]	AX: 6789H, IP: 011FH
MOV AX, [BX+0002H]	Ax: 6789H, IP:0122H

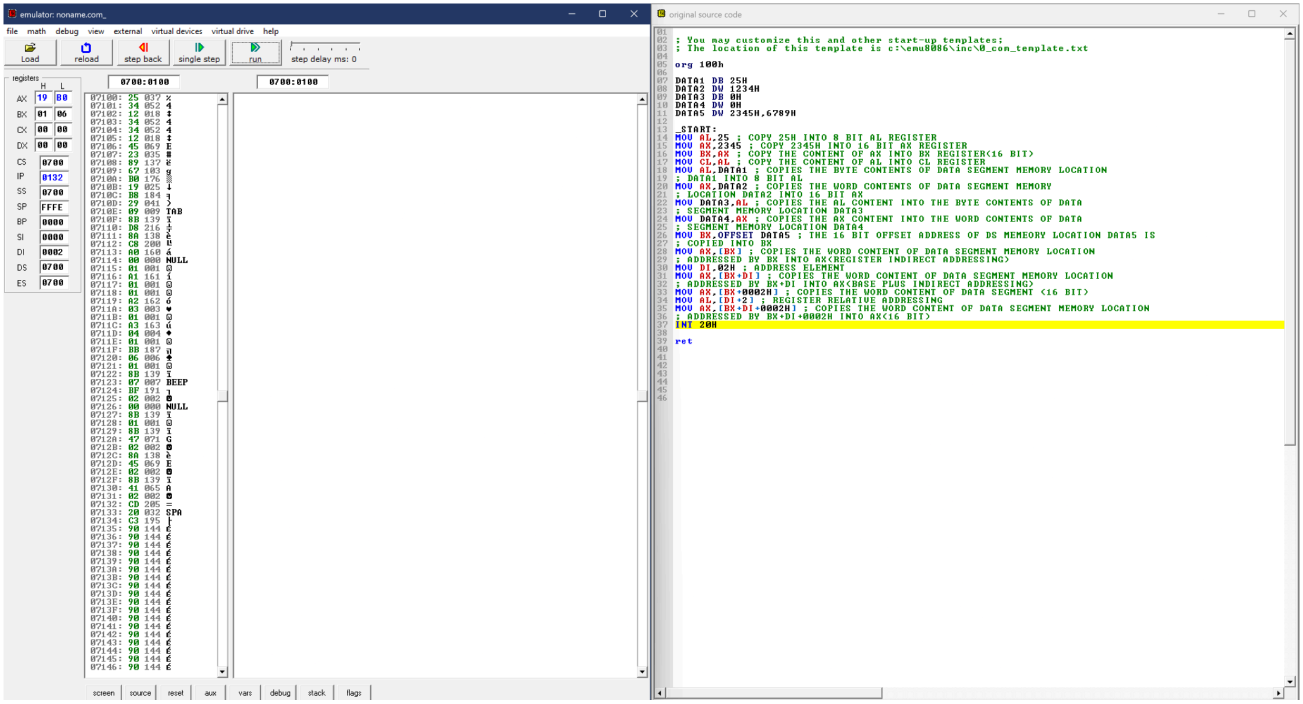
MOV AL, [DI+2]	AX: 6700H, IP: 0125H
MOV AX, [BX+DI+0002H]	AX: 9090H, 0128H
INT 20H	CS: F400H, IP: 0150H, SP FFF8H

Activity #2

Instruction	Register value after the instruction is executed
MOV BX, OFFSET SRC	BX = OFFSET SRC
MOV DI, 0000H	DI = 0000H
MOV AX, [BX+DI+0002H]	AX = 2222H
MOV BX, OFFSET DST	BX = OFFSET DST
MOV SI, 0000H	SI = 0000H
MOV [BX+SI+0000H], AX	no change
MOV BX, OFFSET SRC	BX = OFFSET SRC
MOV DI, 0002H	DI = 0002H
MOV AX, [BX+DI+0002H]	AX = 3333H
MOV BX, OFFSET DST	BX = OFFSET DST
MOV SI, 0002H	SI = 0002H
MOV [BX+SI+0000H], AX	no change
MOV BX, OFFSET SRC	BX = OFFSET SRC
MOV DI, 0004H	DI = 0004H
MOV AX, [BX+DI+0002H]	AX = 4444H
MOV BX, OFFSET DST	BX = OFFSET DST
MOV SI, 0004H	SI = 0004H
MOV [BX+SI+0000H], AX	no change
MOV BX, OFFSET SRC	BX = OFFSET SRC

MOV DI, 0006H	DI = 0006H
MOV AX, [BX+DI+0002H]	AX = 5555H
MOV BX, OFFSET DST	BX = OFFSET DST
MOV SI, 0006H	SI = 0006H
MOV [BX+SI+0000H], AX	no change
INT 20H	no change

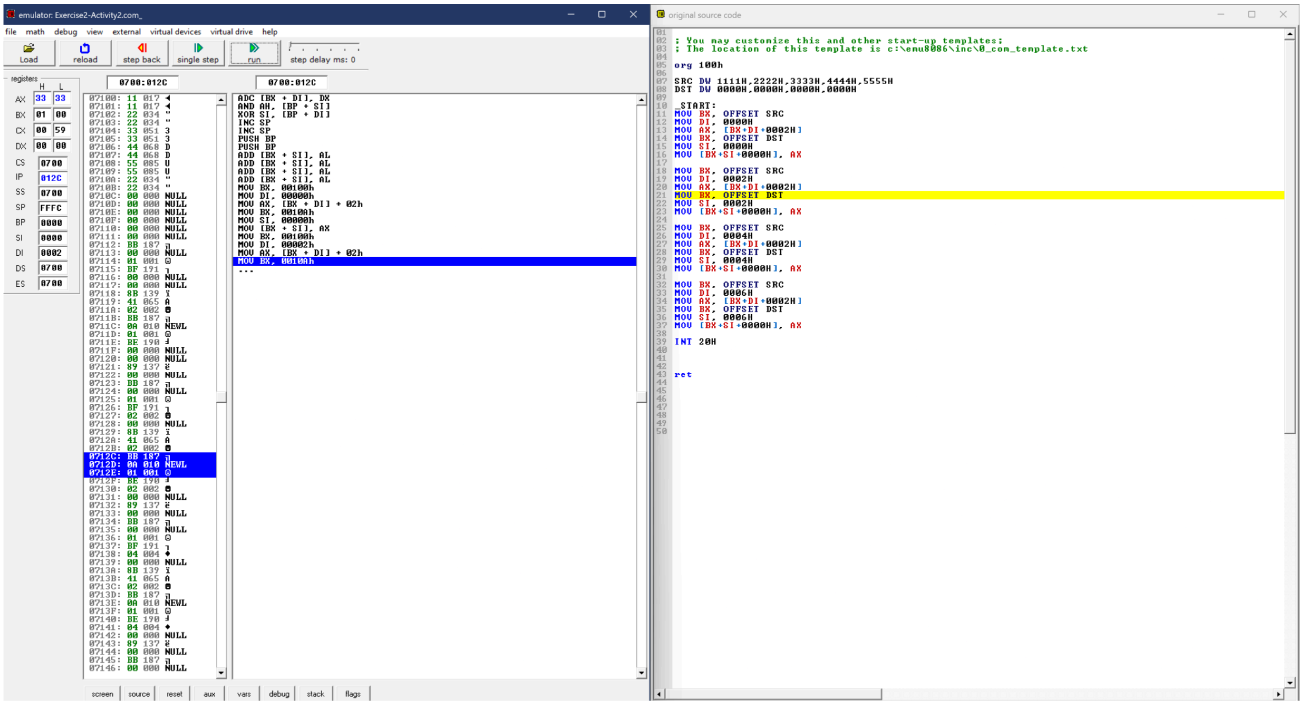
Output Screenshots:



```
     H  L
AX   19 B0
BX   01 06
CX   00 00
DX   00 00
CS   0700
IP   0132
SS   0700
SP   FFFE
BP   0000
SI   0000
DI   0002
DS   0700
ES   0700
```

```
07100: 25 037 %
07101: 34 052 4
07102: 12 018 ↕
07103: 34 052 4
07104: 34 052 4
07105: 12 018 ↕
07106: 45 069 E
07107: 23 035 #
07108: 89 137 ë
07109: 67 103 g
0710A: B0 176 ░
0710B: 19 025 ↓
0710C: B8 184 ╕
0710D: 29 041 )
0710E: 09 009 TAB
0710F: 8B 139 ï
07110: D8 216 ╪
07111: 8A 138 è
07112: C8 200 ╚
07113: A0 160 á
07114: 00 000 NULL
07115: 01 001 ☺
07116: A1 161 í
07117: 01 001 ☺
07118: 01 001 ☺
07119: A2 162 ó
0711A: 03 003 ♥
0711B: 01 001 ☺
0711C: A3 163 ú
0711D: 04 004 ♦
0711E: 01 001 ☺
0711F: BB 187 ╗
07120: 06 006 ♠
07121: 01 001 ☺
07122: 8B 139 ï
07123: 07 007 BEEP
07124: BF 191 ┐
07125: 02 002 ☻
07126: 00 000 NULL
07127: 8B 139 ï
07128: 01 001 ☺
07129: 8B 139 ï
0712A: 47 071 G
0712B: 02 002 ☻
0712C: 8A 138 è
0712D: 45 069 E
0712E: 02 002 ☻
0712F: 8B 139 ï
07130: 41 065 A
07131: 02 002 ☻
07132: CD 205 ═
07133: 20 032 SPA
07134: C3 195 ├
07135: 90 144 É
07136: 90 144 É
07137: 90 144 É
07138: 90 144 É
07139: 90 144 É
0713A: 90 144 É
0713B: 90 144 É
0713C: 90 144 É
0713D: 90 144 É
0713E: 90 144 É
0713F: 90 144 É
07140: 90 144 É
07141: 90 144 É
07142: 90 144 É
07143: 90 144 É
07144: 90 144 É
07145: 90 144 É
07146: 90 144 É
```

```

; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

DATA1 DB 25H
DATA2 DW 1234H
DATA3 DB 0H
DATA4 DW 0H
DATA5 DW 2345H,6789H

_START:
MOV AL,25 ; COPY 25H INTO 8 BIT AL REGISTER
MOV AX,2345 ; COPY 2345H INTO 16 BIT AX REGISTER
MOV BX,AX ; COPY THE CONTENT OF AX INTO BX REGISTER(16 BIT)
MOV CL,AL ; COPY THE CONTENT OF AL INTO CL REGISTER
MOV AL,DATA1 ; COPIES THE BYTE CONTENTS OF DATA SEGMENT MEMORY LOCATION
; DATA1 INTO 8 BIT AL
MOV AX,DATA2 ; COPIES THE WORD CONTENTS OF DATA SEGMENT MEMORY
; LOCATION DATA2 INTO 16 BIT AX
MOV DATA3,AL ; COPIES THE AL CONTENT INTO THE BYTE CONTENTS OF DATA
; SEGMENT MEMORY LOCATION DATA3
MOV DATA4,AX ; COPIES THE AX CONTENT INTO THE WORD CONTENTS OF DATA
; SEGMENT MEMORY LOCATION DATA4
MOV BX,OFFSET DATA5 ; THE 16 BIT OFFSET ADDRESS OF DS MEMEORY LOCATION DATA5 IS
; COPIED INTO BX
MOV AX,[BX] ; COPIES THE WORD CONTENT OF DATA SEGMENT MEMORY LOCATION
; ADDRESSED BY BX INTO AX(REGISTER INDIRECT ADDRESSING)
MOV DI,02H ; ADDRESS ELEMENT
MOV AX,[BX+DI] ; COPIES THE WORD CONTENT OF DATA SEGMENT MEMORY LOCATION
; ADDRESSED BY BX+DI INTO AX(BASE PLUS INDIRECT ADDRESSING)
MOV AX,[BX+0002H] ; COPIES THE WORD CONTENT OF DATA SEGMENT (16 BIT)
MOV AL,[DI+2] ; REGISTER RELATIVE ADDRESSING
MOV AX,[BX+DI+0002H] ; COPIES THE WORD CONTENT OF DATA SEGMENT MEMORY LOCATION
; ADDRESSED BY BX+DI+0002H INTO AX(16 BIT)
INT 20H

ret
```



```
     H   L
AX   33  33
BX   01  00
CX   00  59
DX   00  00
CS   0700
IP   012C
SS   0700
SP   FFFC
BP   0000
SI   0000
DI   0002
DS   0700
ES   0700
```

```
0700:012C
07100: 11 017 ◄
07101: 11 017 ◄
07102: 22 034 "
07103: 22 034 "
07104: 33 051 3
07105: 33 051 3
07106: 44 068 D
07107: 44 068 D
07108: 55 085 U
07109: 55 085 U
0710A: 22 034 "
0710B: 22 034 "
0710C: 00 000 NULL
0710D: 00 000 NULL
0710E: 00 000 NULL
0710F: 00 000 NULL
07110: 00 000 NULL
07111: 00 000 NULL
07112: BB 187 ╗
07113: 00 000 NULL
07114: 01 001 ☺
07115: BF 191 ┐
07116: 00 000 NULL
07117: 00 000 NULL
07118: 8B 139 ï
07119: 41 065 A
0711A: 02 002 ☻
0711B: BB 187 ╗
0711C: 0A 010 NEWL
0711D: 01 001 ☺
0711E: BE 190 ╛
0711F: 00 000 NULL
07120: 00 000 NULL
07121: 89 137 ë
07122: 00 000 NULL
07123: BB 187 ╗
07124: 00 000 NULL
07125: 01 001 ☺
07126: BF 191 ┐
07127: 02 002 ☻
07128: 00 000 NULL
07129: 8B 139 ï
0712A: 41 065 A
0712B: 02 002 ☻
0712C: BB 187 ╗
0712D: 0A 010 NEWL
0712E: 01 001 ☺
0712F: BE 190 ╛
07130: 02 002 ☻
07131: 00 000 NULL
07132: 89 137 ë
07133: 00 000 NULL
07134: BB 187 ╗
07135: 00 000 NULL
07136: 01 001 ☺
07137: BF 191 ┐
07138: 04 004 ♦
07139: 00 000 NULL
0713A: 8B 139 ï
0713B: 41 065 A
0713C: 02 002 ☻
0713D: BB 187 ╗
0713E: 0A 010 NEWL
0713F: 01 001 ☺
07140: BE 190 ╛
07141: 04 004 ♦
07142: 00 000 NULL
07143: 89 137 ë
07144: 00 000 NULL
07145: BB 187 ╗
07146: 00 000 NULL
```

```
0700:012C
ADC [BX + DI], DX
AND AH, [BP + SI]
XOR SI, [BP + DI]
INC SP
INC SP
PUSH BP
PUSH BP
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
MOV BX, 00100h
MOV DI, 00000h
MOV AX, [BX + DI] + 02h
MOV BX, 0010Ah
MOV SI, 00000h
MOV [BX + SI], AX
MOV BX, 00100h
MOV DI, 00002h
MOV AX, [BX + DI] + 02h
MOV BX, 0010Ah
...
```

```

; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

SRC DW 1111H,2222H,3333H,4444H,5555H
DST DW 0000H,0000H,0000H,0000H

_START:
MOV BX, OFFSET SRC
MOV DI, 0000H
MOV AX, [BX+DI+0002H]
MOV BX, OFFSET DST
MOV SI, 0000H
MOV [BX+SI+0000H], AX

MOV BX, OFFSET SRC
MOV DI, 0002H
MOV AX, [BX+DI+0002H]
MOV BX, OFFSET DST
MOV SI, 0002H
MOV [BX+SI+0000H], AX

MOV BX, OFFSET SRC
MOV DI, 0004H
MOV AX, [BX+DI+0002H]
MOV BX, OFFSET DST
MOV SI, 0004H
MOV [BX+SI+0000H], AX

MOV BX, OFFSET SRC
MOV DI, 0006H
MOV AX, [BX+DI+0002H]
MOV BX, OFFSET DST
MOV SI, 0006H
MOV [BX+SI+0000H], AX

INT 20H



ret
```