Name : Kreena Shah
Div / Batch : C32
Sapid : 60004210243
Subject : Processor Organization and Architecture (POA)

Expermiment 5

Aim : Assembly program for 16-bit Addition / Subtraction using Direct, Immediate and Register Addressing Mode.

Direct Addressing Mode

Theory :
In this type of addressing mode the effective address is directly given in the instruction as displacement.
Example:
MOV AX, [DISP]
MOV AX, [0500]

Code :

Addition

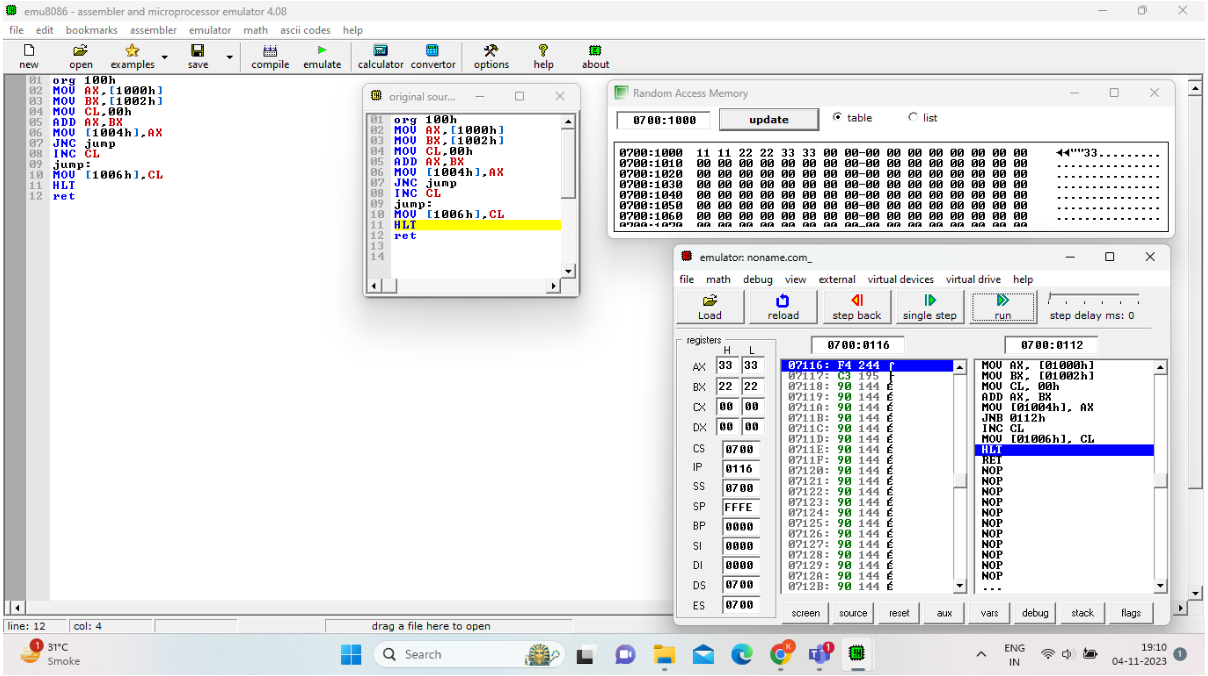
```
org 100h
MOV AX,[1000h]
MOV BX,[1002h]
MOV CL,00h
ADD AX,BX
MOV [1004h],AX
JNC jump
INC CL
jump:
MOV [1006h],CL
HLT
ret
```

Subtraction

```
org 100h
MOV AX,[1000h]
MOV BX,[1002h]
MOV CL,00h
SUB AX,BX
MOV[1004h],AX
JNC jump
INC CL
NOT AX
ADD AX,0001h
jump:
MOV [1006h],AX
HLT
Ret
```

Screenshots :

Addition


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
original sour...
Random Access Memory
update
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run
screen source reset aux vars debug stack flags
line: 12 col: 4
drag a file here to open
31°C Smoke
Search
ENG IN 19:10 04-11-2023

```
org 100h
MOV AX,[1000h]
MOV BX,[1002h]
MOV CL,00h
ADD AX,BX
MOV [1004h],AX
JNC jump
INC CL
jump:
MOV [1006h],CL
HLT
ret
```

0700:1000 ● table ○ list

```
0700:1000  11 11 22 22 33 33 00 00-00 00 00 00 00 00 00 00   ◄◄""33..........
0700:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

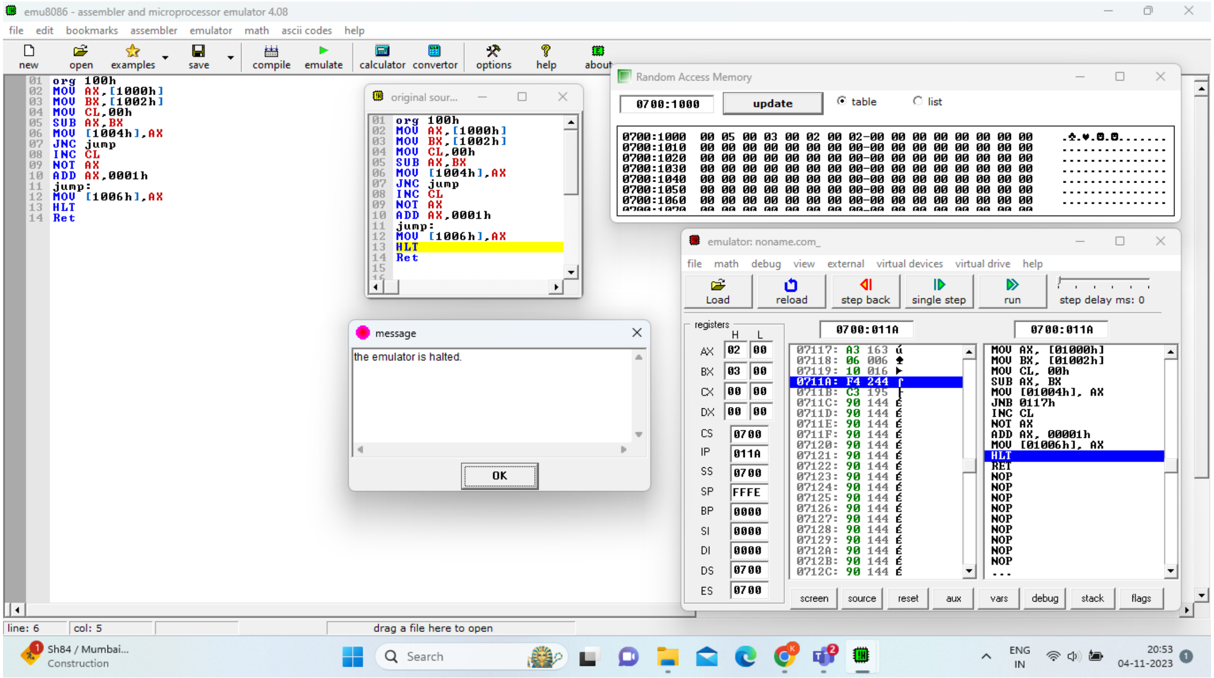
step delay ms: 0

```
registers
     H   L
AX   33  33
BX   22  22
CX   00  00
DX   00  00
CS   0700
IP   0116
SS   0700
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0700
ES   0700
```

```
0700:0116
07116: F4 244 ⌠
07117: C3 195 ├
07118: 90 144 É
07119: 90 144 É
0711A: 90 144 É
0711B: 90 144 É
0711C: 90 144 É
0711D: 90 144 É
0711E: 90 144 É
0711F: 90 144 É
07120: 90 144 É
07121: 90 144 É
07122: 90 144 É
07123: 90 144 É
07124: 90 144 É
07125: 90 144 É
07126: 90 144 É
07127: 90 144 É
07128: 90 144 É
07129: 90 144 É
0712A: 90 144 É
0712B: 90 144 É
```

```
0700:0112
MOV AX, [01000h]
MOV BX, [01002h]
MOV CL, 00h
ADD AX, BX
MOV [01004h], AX
JNB 0112h
INC CL
MOV [01006h], CL
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

Subtraction


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
original sour...
Random Access Memory
update
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run
screen source reset aux vars debug stack flags
message
OK
line: 6 col: 5
drag a file here to open
Sh84 / Mumbai... Construction
Search
ENG IN 20:53 04-11-2023

```
org 100h
MOV AX,[1000h]
MOV BX,[1002h]
MOV CL,00h
SUB AX,BX
MOV [1004h],AX
JNC jump
INC CL
NOT AX
ADD AX,0001h
jump:
MOV [1006h],AX
HLT
Ret
```

the emulator is halted.

0700:1000 ● table ○ list

```
0700:1000  00 05 00 03 00 02 00 02-00 00 00 00 00 00 00 00   .♣.♥.☻.☻........
0700:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:1060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

step delay ms: 0

```
registers
     H   L
AX   02  00
BX   03  00
CX   00  00
DX   00  00
CS   0700
IP   011A
SS   0700
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0700
ES   0700
```

```
0700:011A
07117: A3 163 ú
07118: 06 006 ♠
07119: 10 016 ►
0711A: F4 244 ⌠
0711B: C3 195 ├
0711C: 90 144 É
0711D: 90 144 É
0711E: 90 144 É
0711F: 90 144 É
07120: 90 144 É
07121: 90 144 É
07122: 90 144 É
07123: 90 144 É
07124: 90 144 É
07125: 90 144 É
07126: 90 144 É
07127: 90 144 É
07128: 90 144 É
07129: 90 144 É
0712A: 90 144 É
0712B: 90 144 É
0712C: 90 144 É
```

```
0700:011A
MOV AX, [01000h]
MOV BX, [01002h]
MOV CL, 00h
SUB AX, BX
MOV [01004h], AX
JNB 0117h
INC CL
NOT AX
ADD AX, 00001h
MOV [01006h], AX
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

Indirect Addressing

Theory :
In this type of addressing mode the source operand is a 8 bit or 16 bit data. Destination operand can never be immediate data.
Example:
MOV AX, 2000
MOV CL, 0A
ADD AL, 45
AND AX, 0000

Code :

Addition

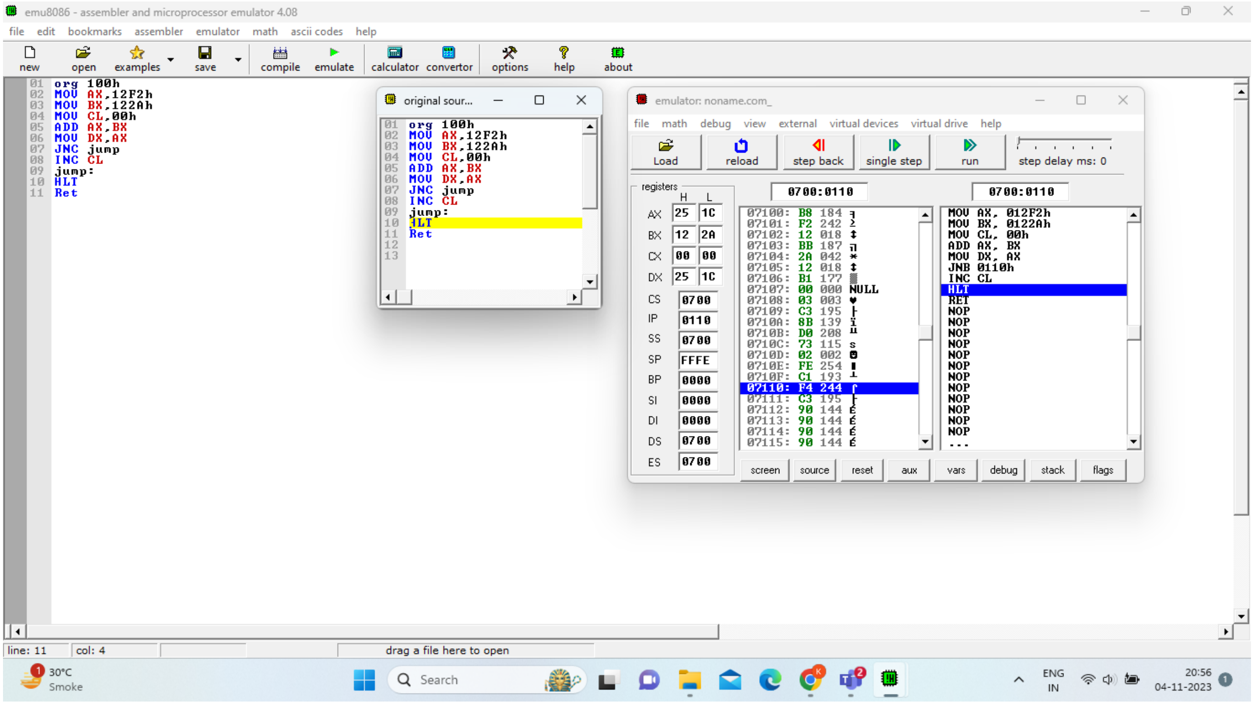
```
org 100h
MOV AX,12F2h
MOV BX,122Ah
MOV CL,00h
ADD AX,BX
MOV DX,AX
JNC jump
INC CL
jump:
HLT
Ret
```

Subtraction

```
org 100h
MOV AX,5656H
MOV BX,345FH
MOV CL,00h
SUB AX,BX
MOV DX,AX
JNC jump
INC CL
NOT AX
ADD AX,0001h
jump:
HLT
ret
```

Screenshots :

Addition


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
original sour...
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
screen source reset aux vars debug stack flags
line: 11 col: 4
drag a file here to open
30°C Smoke
Search
ENG IN
20:56 04-11-2023

```
org 100h
MOV AX,12F2h
MOV BX,122Ah
MOV CL,00h
ADD AX,BX
MOV DX,AX
JNC jump
INC CL
jump:
HLT
Ret
```

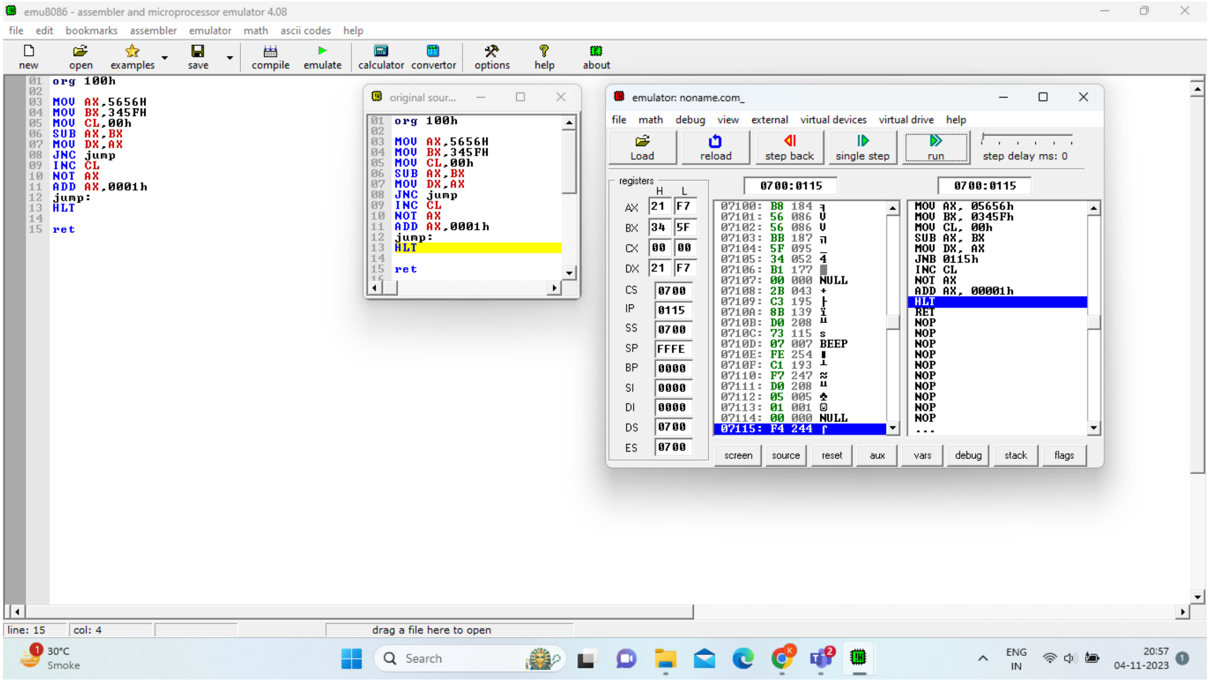
```
org 100h
MOV AX,12F2h
MOV BX,122Ah
MOV CL,00h
ADD AX,BX
MOV DX,AX
JNC jump
INC CL
jump:
HLT
Ret
```

```
    H  L
AX  25 1C
BX  12 2A
CX  00 00
DX  25 1C
CS  0700
IP  0110
SS  0700
SP  FFFE
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700
```

```
0700:0110
07100: B8 184 ╕
07101: F2 242 ≥
07102: 12 018 ↕
07103: BB 187 ╗
07104: 2A 042 *
07105: 12 018 ↕
07106: B1 177 ▒
07107: 00 000 NULL
07108: 03 003 ♥
07109: C3 195 ├
0710A: 8B 139 ï
0710B: D0 208 ╨
0710C: 73 115 s
0710D: 02 002 ☻
0710E: FE 254 ■
0710F: C1 193 ┴
07110: F4 244 ⌠
07111: C3 195 ├
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

```
0700:0110
MOV AX, 012F2h
MOV BX, 0122Ah
MOV CL, 00h
ADD AX, BX
MOV DX, AX
JNB 0110h
INC CL
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

Subtraction


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
screen source reset aux vars debug stack flags
line: 15 col: 4 drag a file here to open
30°C Smoke Search ENG IN 20:57 04-11-2023

```
org 100h

MOV AX,5656H
MOV BX,345FH
MOV CL,00h
SUB AX,BX
MOV DX,AX
JNC jump
INC CL
NOT AX
ADD AX,0001h
jump:
HLT

ret
```

original sour...

```
org 100h

MOV AX,5656H
MOV BX,345FH
MOV CL,00h
SUB AX,BX
MOV DX,AX
JNC jump
INC CL
NOT AX
ADD AX,0001h
jump:
HLT

ret
```

```
registers
     H  L
AX  21 F7
BX  34 5F
CX  00 00
DX  21 F7
CS  0700
IP  0115
SS  0700
SP  FFFE
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700
```

```
0700:0115
07100: B8 184 ╕
07101: 56 086 V
07102: 56 086 V
07103: BB 187 ╗
07104: 5F 095 _
07105: 34 052 4
07106: B1 177 ▒
07107: 00 000 NULL
07108: 2B 043 +
07109: C3 195 ├
0710A: 8B 139 ï
0710B: D0 208 ╨
0710C: 73 115 s
0710D: 07 007 BEEP
0710E: FE 254 ■
0710F: C1 193 ┴
07110: F7 247 ≈
07111: D0 208 ╨
07112: 05 005 ♣
07113: 01 001 ☺
07114: 00 000 NULL
07115: F4 244 ⌠
```

```
0700:0115
MOV AX, 05656h
MOV BX, 0345Fh
MOV CL, 00h
SUB AX, BX
MOV DX, AX
JNB 0115h
INC CL
NOT AX
ADD AX, 00001h
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

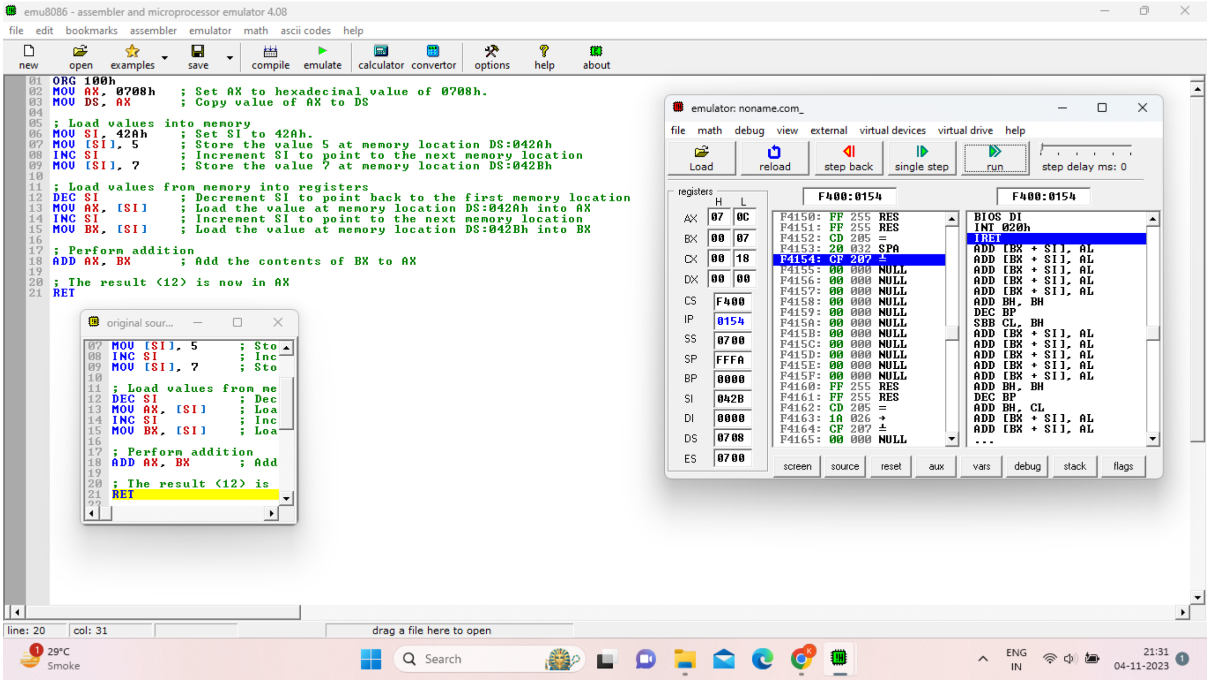
Register Addressing

Theory :
In this type of addressing mode both the operands are registers.
Example:
MOV AX, BX
XOR AX, DX
ADD AL, BL

Code :

```
ORG 100h
MOV AX, 0708h
MOV DS, AX
MOV SI, 42Ah
MOV [SI], 5
INC SI
MOV [SI], 7
DEC SI
MOV AX, [SI]
INC SI
MOV BX, [SI]
ADD AX, BX
RET
```

Screenshots :


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
screen source reset aux vars debug stack flags
line: 20 col: 31 drag a file here to open

```
ORG 100h
MOV AX, 0708h    ; Set AX to hexadecimal value of 0708h.
MOV DS, AX       ; Copy value of AX to DS

; Load values into memory
MOV SI, 42Ah     ; Set SI to 42Ah.
MOV [SI], 5      ; Store the value 5 at memory location DS:042Ah
INC SI           ; Increment SI to point to the next memory location
MOV [SI], 7      ; Store the value 7 at memory location DS:042Bh

; Load values from memory into registers
DEC SI           ; Decrement SI to point back to the first memory location
MOV AX, [SI]     ; Load the value at memory location DS:042Ah into AX
INC SI           ; Increment SI to point to the next memory location
MOV BX, [SI]     ; Load the value at memory location DS:042Bh into BX

; Perform addition
ADD AX, BX       ; Add the contents of BX to AX

; The result (12) is now in AX
RET
```

registers

	H	L
AX	07	0C
BX	00	07
CX	00	18
DX	00	00

CS	F400
IP	0154
SS	0700
SP	FFFA
BP	0000
SI	042B
DI	0000
DS	0708
ES	0700

```
F400:0154

F4150: FF 255 RES
F4151: FF 255 RES
F4152: CD 205 =
F4153: 20 032 SPA
F4154: CF 207 ±
F4155: 00 000 NULL
F4156: 00 000 NULL
F4157: 00 000 NULL
F4158: 00 000 NULL
F4159: 00 000 NULL
F415A: 00 000 NULL
F415B: 00 000 NULL
F415C: 00 000 NULL
F415D: 00 000 NULL
F415E: 00 000 NULL
F415F: 00 000 NULL
F4160: FF 255 RES
F4161: FF 255 RES
F4162: CD 205 =
F4163: 1A 026 →
F4164: CF 207 ±
F4165: 00 000 NULL
```

```
F400:0154

BIOS DI
INT 020h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
SBB CL, BH
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADD BH, CL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

Conclusion :

Addressing modes are critical for optimizing code and enhancing program efficiency. They enable programmers to choose the most suitable method for data access and manipulation. Mastering these modes empowers programmers to write efficient and maintainable assembly code, making the 8086 microprocessor a powerful tool for various applications.