

## 1. $DX = A + B + C$

The screenshot displays an x86 assembly editor and emulator. The assembly code is as follows:

```
01  ORG 100H
02  .DATA ; Data segment starts
03
04  A DW 9
05  B DW 4
06  C DW 5
07
08
09  .CODE ; CODE SEGMENT HERE
10  MAIN PROC ; INITIALIZE DATA SEGEMENT
11  MOV AX, @DATA
12  MOV DS, AX
13
14
15  XOR DX, DX
16  MOV DX, A
17  ADD DX, B
18  ADD DX, C
19
20  MAIN ENDP; END PROCEDURE
21  END MAIN
22  ret
23
24
25
26
```

The emulator window shows the registers and memory. The registers are:

Register	H	L
AX	07	00
BX	00	00
CX	00	1A
DX	00	12
CS	07	00
IP	01	2E
SS	07	00
SP	FF	FE
BP	00	00
SI	00	00
DI	00	00
DS	07	00
ES	07	00

The extended value viewer shows the value of DX in various formats:

Format	Value
hex	00 12
bin	00000000 00010010
oct	000 022
decimal 8 bit unsigned	18
signed	18
decimal 16 bit unsigned	18
signed	18

A message box indicates that the emulator is halted.

## 2. $BX = A + D - B - C$

The screenshot displays an x86 assembly editor and emulator. The assembly code is as follows:

```
01  ORG 100H
02  .DATA ; Data segment starts
03
04  A DW 9
05  B DW 4
06  C DW 5
07  D DW 8
08
09
10  .CODE ; CODE SEGMENT HERE
11  MAIN PROC ; INITIALIZE DATA SEGEMENT
12  MOV AX, @DATA
13  MOV DS, AX
14
15  XOR BX, BX
16  MOV BX, A
17  ADD BX, D
18  SUB BX, B
19  SUB BX, C
20
21  MAIN ENDP; END PROCEDURE
22  END MAIN
23  ret
24
25
26
27
```

The emulator window shows the registers and memory. The registers are:

Register	H	L
AX	07	00
BX	00	08
CX	00	20
DX	00	00
CS	07	00
IP	01	34
SS	07	00
SP	FF	FE
BP	00	00
SI	00	00
DI	00	00
DS	07	00
ES	07	00

The extended value viewer shows the value of BX in various formats:

Format	Value
hex	00 08
bin	00000000 00001000
oct	000 010
decimal 8 bit unsigned	8
signed	8
decimal 16 bit unsigned	8
signed	8

A message box indicates that the emulator is halted.

### 3. Initialize the BX to 5H. Evaluate BX - E + 10H and Store Result into AX.

The screenshot displays an x86 assembler and emulator environment. The main window shows the assembly code for a program. The code is as follows:

```
01 ORG 100H
02 .DATA ; Data segment starts
03
04 E DW 2H
05
06 .CODE ; CODE SEGMENT HERE
07 MAIN PROC ; INITIALIZE DATA SEGEMENT
08     MOV AX, @DATA
09     MOV DS, AX
10
11     MOV BX, 5H
12     SUB BX, E
13     ADD BX, 10H
14
15     MOV AX, BX
16     XOR BX, BX
17
18     MAIN ENDP; END PROCEDURE
19 END MAIN
20 ret
21
22
23
24
```

The registers window shows the current state of the registers:

Register	H	L
AX	00	13
BX	00	00
CX	00	16
DX	00	00
CS	0700	
IP	012A	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

The extended value viewer shows the value of the BX register:

Watch	Word	Byte
hex:	00	00
bin:	00000000	00000000
oct:	000	000
decimal 8 bit:	0	0
signed:	0	0
ascii:		
decimal 16 bit:	0	0
signed:	0	0

A message box is displayed, stating: "the emulator is halted."