

Variables

Variable is a memory location. For a programmer it is much easier to have some value be kept in a variable named "**var1**" then at the address 5A73:235B, especially when you have 10 or more variables.

8086 compiler supports two types of variables: **BYTE** and **WORD**.

Syntax for a variable declaration:

name **DB** *value*

name **DW** *value*

DB - stays for Define Byte.

DW - stays for Define Word.

name - can be any letter or digit combination, though it should start with a letter. It's possible to declare unnamed variables by not specifying the name (this variable will have an address but no name).

value - can be any numeric value in any supported numbering system (hexadecimal, binary, or decimal), or "?" symbol for variables that are not initialized.

Ex/write program to load (AL)register with value (07) stored in variable and copy the value (1234h) that is stored in another variable to (BX) register.

The screenshot shows an 8086 emulator window titled "emulator: noname.com". The main window displays assembly code and registers. The assembly code window shows the following code:

```
01 ORG 100h
02
03 MOV AL, var1
04 MOV BX, var2
05
06 RET ; stops the program.
07
08 VAR1 DB 7
09 var2 DW 1234h
10
11
12
13
```

The registers window shows the following values:

Register	H	L
AX	00	07
BX	12	34
CX	00	0B
DX	00	00
CS	0700	
IP	0107	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

The assembly code window also shows the following code:

```
MOV AL, [00108h]
MOV BX, [00109h]
RET
POP ES
XOR AL, 012h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

A red circle highlights the variable declarations in the assembly code window, and a red arrow points from the circle to the registers window, indicating the relationship between the variables and the registers.

YOUSSEF NAFI