

7) Logical XOR (Exclusive OR) between all bits of two operands. Result is stored in first operand.

These rules apply:

1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 1 = 1
0 XOR 0 = 0

Example:

```
MOV AL, 00000111b
MOV BL, 00001111b
XOR AL, BL
RET
```

8) Unsigned multiply.

Algorithm:

when operand is a **byte**:

AX = AL * operand.

when operand is a **word**:

(DX AX) = AX * operand.

Example:

```
MOV AL, 2
MOV BL, 4
MUL BL
RET
```

9) Unsigned divide.

Algorithm:

when operand is a **byte**:

AL = AX / operand

AH = remainder

when operand is a **word**:

AX = (DX AX) / operand

DX = remainder

Example:

```
MOV AX, 203
```

```
MOV BL, 4
```

```
DIV BL
```

```
RET
```

H.W

AL = 15 , BL = 7 , CH = 2

Use the "Logical XOR" between AL and BL, and then divide the result by CH. Show and explain your result by emu8086.

10)Loop

Decrease CX, jump to label if CX not zero.

Algorithm:

- CX = CX - 1
- if CX < 0 then
 - jump
 - else
 - no jump, continue

Example:

```
include 'emu8086.inc'
```

```
ORG 100h  
MOV CX, 5  
label1:  
PRINTN 'loop'  
LOOP label1  
RET
```

Variables

Variable is a memory location.

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.

Example:

```
ORG 100h
MOV  AL, VAR1
MOV  BX, VAR2

ADD  AL , BL

RET
VAR1  DB 2
VAR2  DW 8
END
```