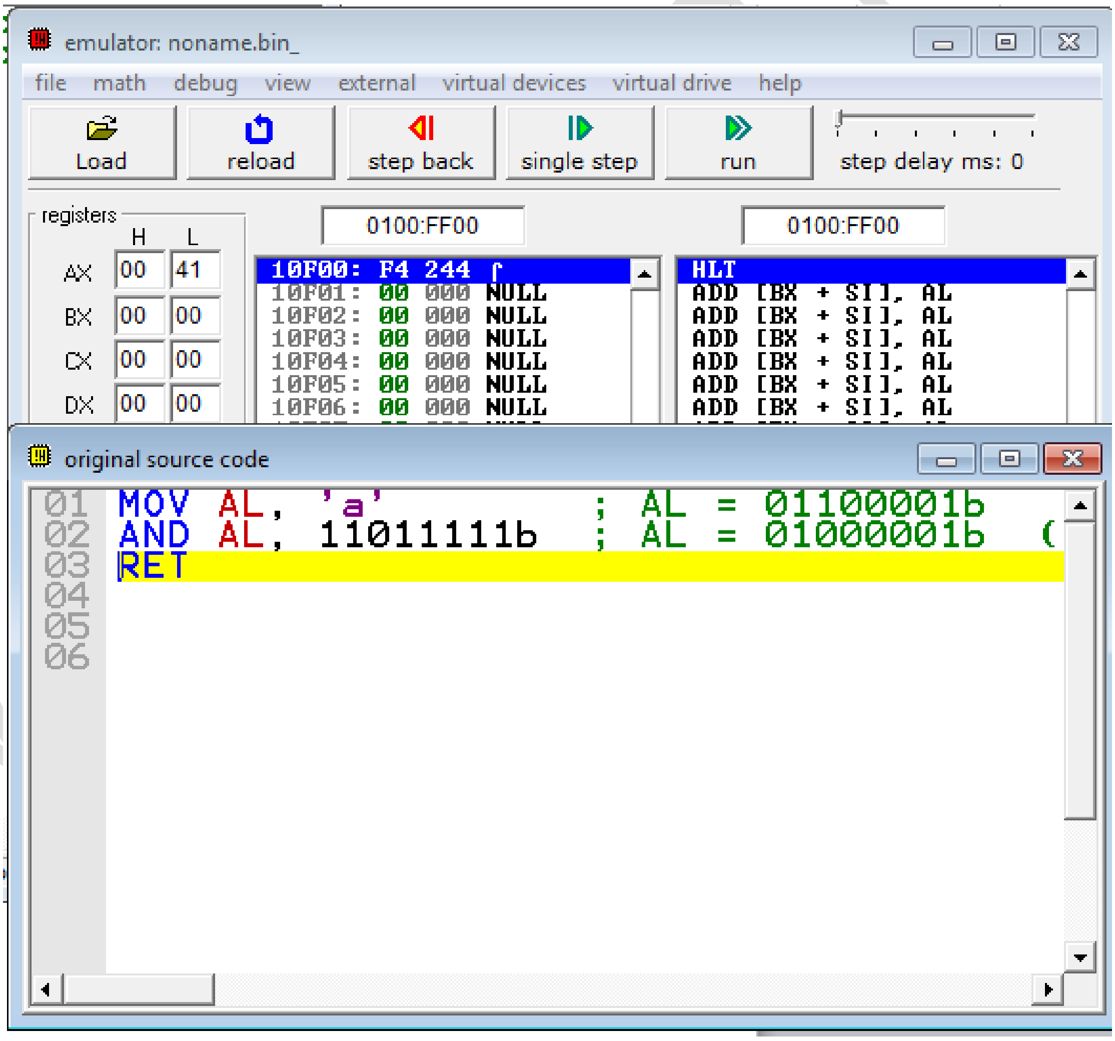
AND	REG, memory memory, REG REG, REG memory, immediate REG, immediate	Logical AND between all bits of two operands. Result is stored in operand1. These rules apply: 1 AND 1 = 1 1 AND 0 = 0 0 AND 1 = 0 0 AND 0 = 0 Example: MOV AL, 'a' ; AL = 01100001b AND AL, 11011111b ; AL = 01000001b ('A') RET C Z S O P 0 r r 0 r



	H	L
AX	00	41
BX	00	00
CX	00	00
DX	00	00

0100:FF00

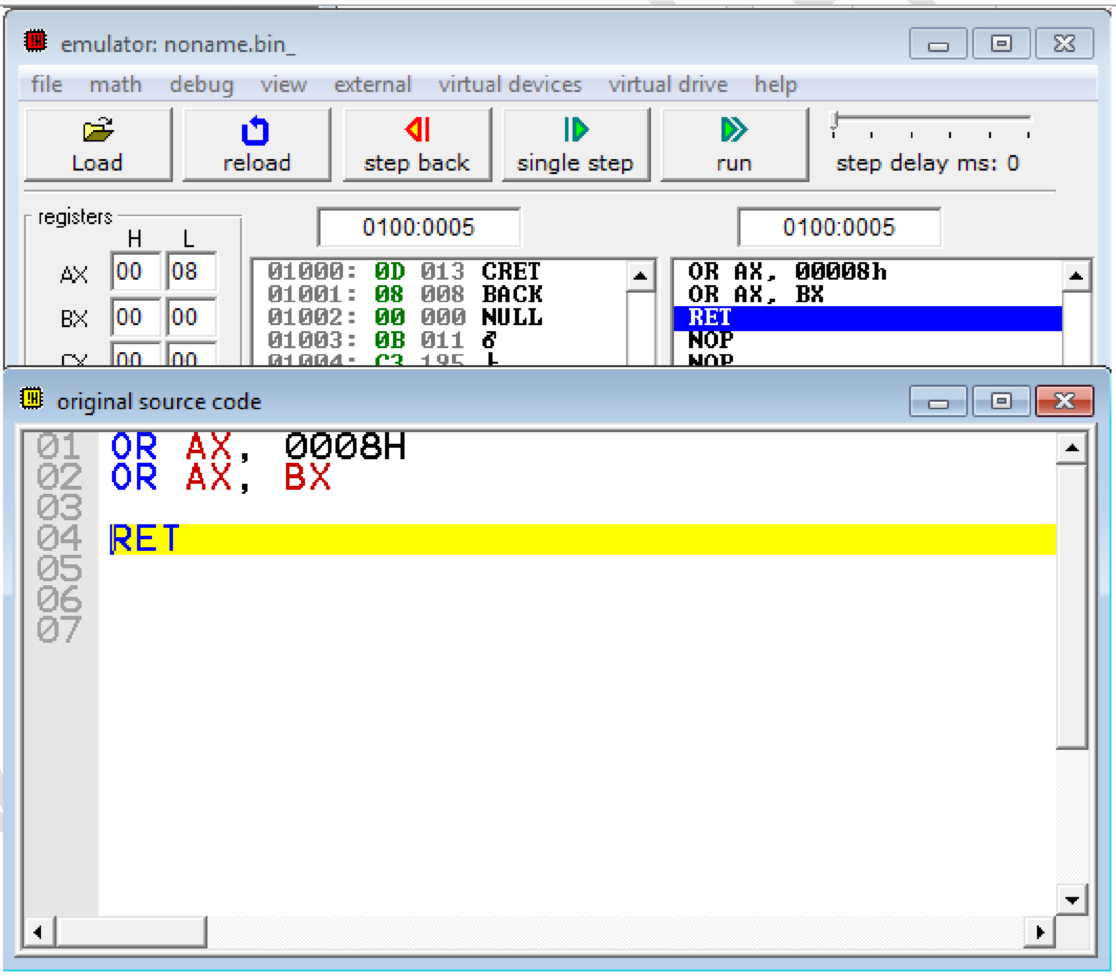
```
10F00: F4 244 ⌠
10F01: 00 000 NULL
10F02: 00 000 NULL
10F03: 00 000 NULL
10F04: 00 000 NULL
10F05: 00 000 NULL
10F06: 00 000 NULL
```

0100:FF00

```
HLT
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
```

```
MOV AL, 'a'          ; AL = 01100001b
AND AL, 11011111b    ; AL = 01000001b   (
RET
```

OR	REG, memory memory, REG REG, REG memory, immediate REG, immediate	Logical OR between all bits of two operands. Result is stored in first operand. These rules apply: 1 OR 1 = 1 1 OR 0 = 1 0 OR 1 = 1 0 OR 0 = 0 Example: MOV AL, 'A' ; AL = 01000001b OR AL, 00100000b ; AL = 01100001b ('a') RET C Z S O P A 0 r r 0 r ?



0100:0005

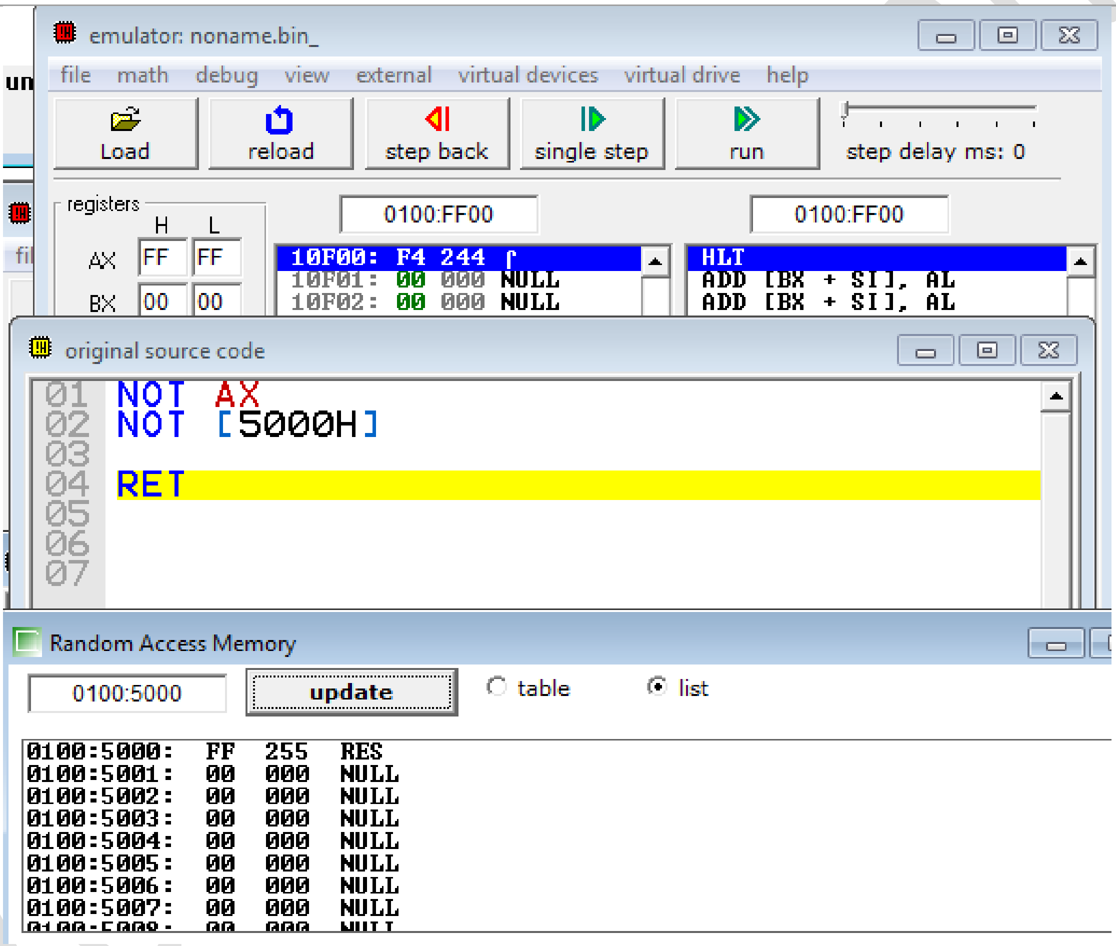
```
01000: 0D 013 CRET
01001: 08 008 BACK
01002: 00 000 NULL
01003: 0B 011 ♂
01004: C3 195 ┤
```

0100:0005

```
OR AX, 0008h
OR AX, BX
RET
NOP
NOP
```

```
OR  AX,  0008H
OR  AX,  BX

RET
```

NOT	REG memory	Invert each bit of the operand. Algorithm: - if bit is 1 turn it to 0. - if bit is 0 turn it to 1. Example: MOV AL, 00011011b NOT AL ; AL = 11100100b RET C Z S O P A unchanged



```
    H  L
AX  FF FF
BX  00 00
```

0100:FF00

```
10F00: F4 244 ſ
10F01: 00 000 NULL
10F02: 00 000 NULL
```

0100:FF00

```
HLT
ADD [BX + SI], AL
ADD [BX + SI], AL
```

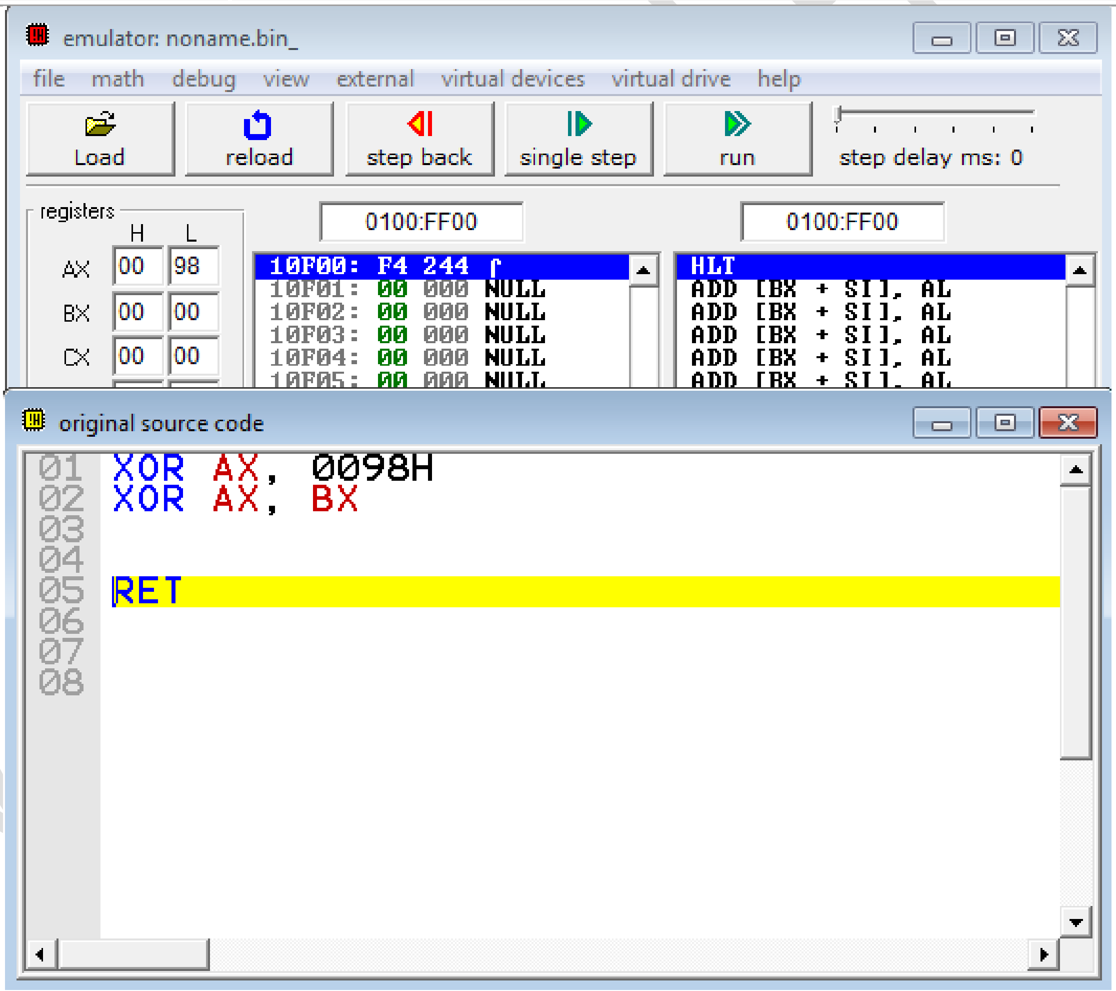
```
NOT AX
NOT [5000H]

RET
```

0100:5000

```
0100:5000:  FF  255  RES
0100:5001:  00  000  NULL
0100:5002:  00  000  NULL
0100:5003:  00  000  NULL
0100:5004:  00  000  NULL
0100:5005:  00  000  NULL
0100:5006:  00  000  NULL
0100:5007:  00  000  NULL
0100:5008:  00  000  NULL
```

Yousif Nihad

XOR	REG, memory memory, REG REG, REG memory, immediate REG, immediate	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 XOR AL, 00000010b ; AL = 00000101b RET C Z S O P A 0 r r 0 r ?

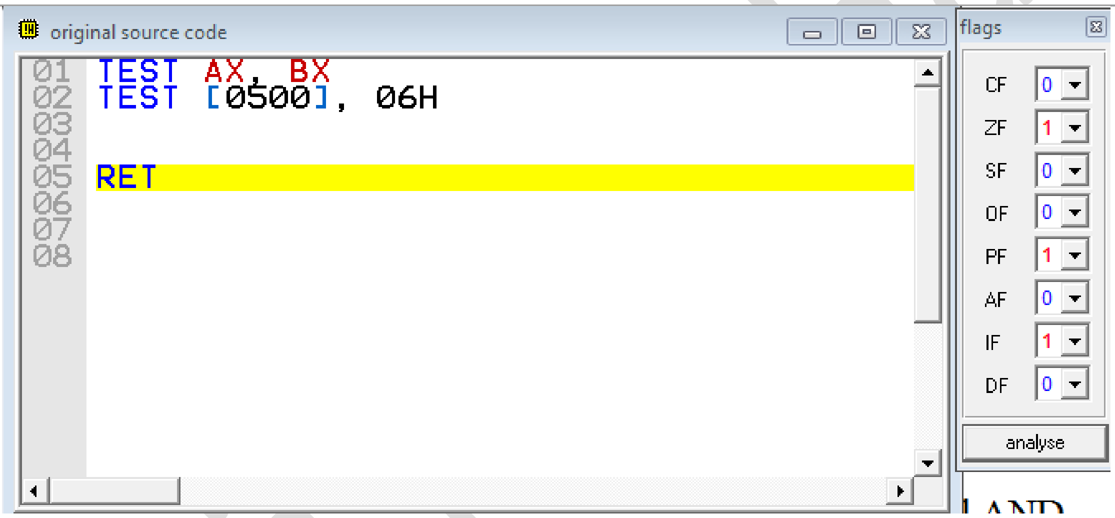


```
AX  00  98
BX  00  00
CX  00  00

0100:FF00                      0100:FF00

10F00:  F4 244 ſ               HLT
10F01:  00 000 NULL            ADD [BX + SI], AL
10F02:  00 000 NULL            ADD [BX + SI], AL
10F03:  00 000 NULL            ADD [BX + SI], AL
10F04:  00 000 NULL            ADD [BX + SI], AL
10F05:  00 000 NULL            ADD [BX + SI], AL
```

```
XOR AX, 0098H
XOR AX, BX


RET
```

TEST	REG, memory memory, REG	Logical AND between all bits of two operands for flags only. These flags are effected: **ZF, SF, PF.** Result is not stored anywhere.

	REG, REG memory, immediate REG, immediate	These rules apply: 1 AND 1 = 1 1 AND 0 = 0 0 AND 1 = 0 0 AND 0 = 0 Example: MOV AL, 00000101b TEST AL, 1 ; ZF = 0. TEST AL, 10b ; ZF = 1. RET C Z S O P 0 r r 0 r

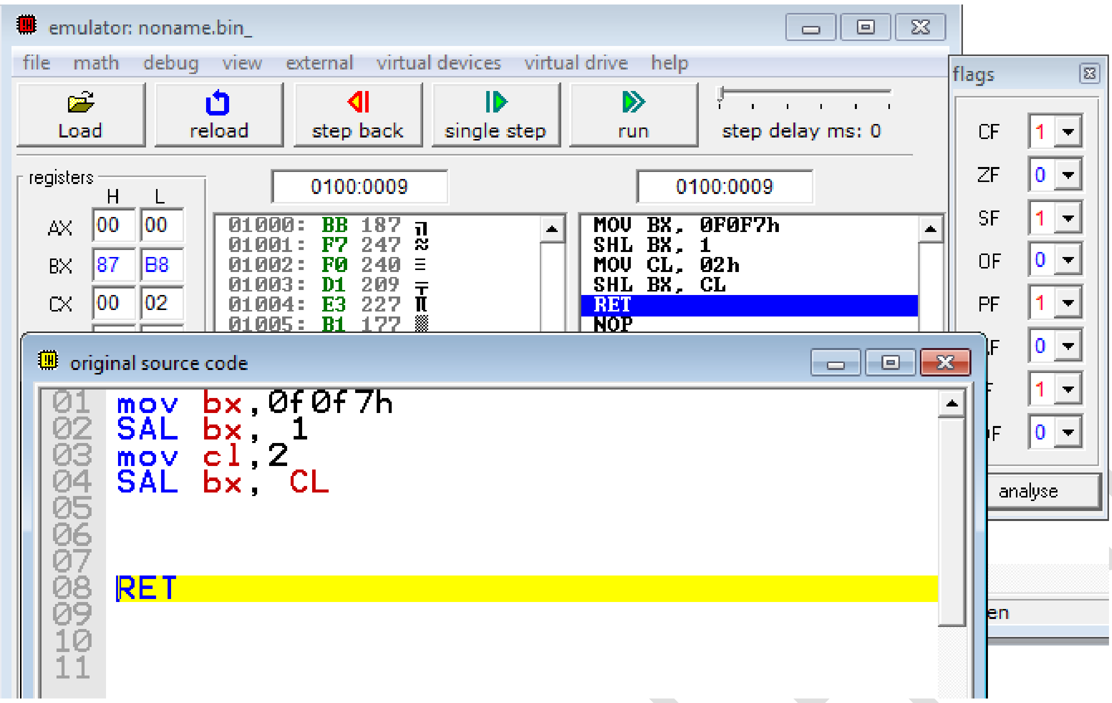


```
TEST AX, BX
TEST [0500], 06H


RET
```

CF 0
ZF 1
SF 0
OF 0
PF 1
AF 0
IF 1
DF 0

SAL	memory, immediate REG, immediate	Shift Arithmetic operand1 Left. The number of shifts is set by operand2. Algorithm:

	memory, CL REG, CL	• Shift all bits left, the bit that goes off is set to CF. • Zero bit is inserted to the right-most position. Example: MOV AL, 0E0h ; AL = 11100000b SAL AL, 1 ; AL = 11000000b, CF=1. RET C O r r OF=0 if first operand keeps original sign.
SHL	memory, immediate REG, immediate memory, CL REG, CL	Shift operand1 Left. The number of shifts is set by operand2. Algorithm: • Shift all bits left, the bit that goes off is set to CF. • Zero bit is inserted to the right-most position. Example: MOV AL, 11100000b SHL AL, 1 ; AL = 11000000b, CF=1. RET C O r r OF=0 if first operand keeps original sign.



registers: AX H 00 L 00; BX H 87 L B8; CX H 00 L 02

0100:0009

```
01000: BB 187 ╗
01001: F7 247 ≈
01002: F0 240 ≡
01003: D1 209 ╤
01004: E3 227 π
01005: B1 177 ▒
```

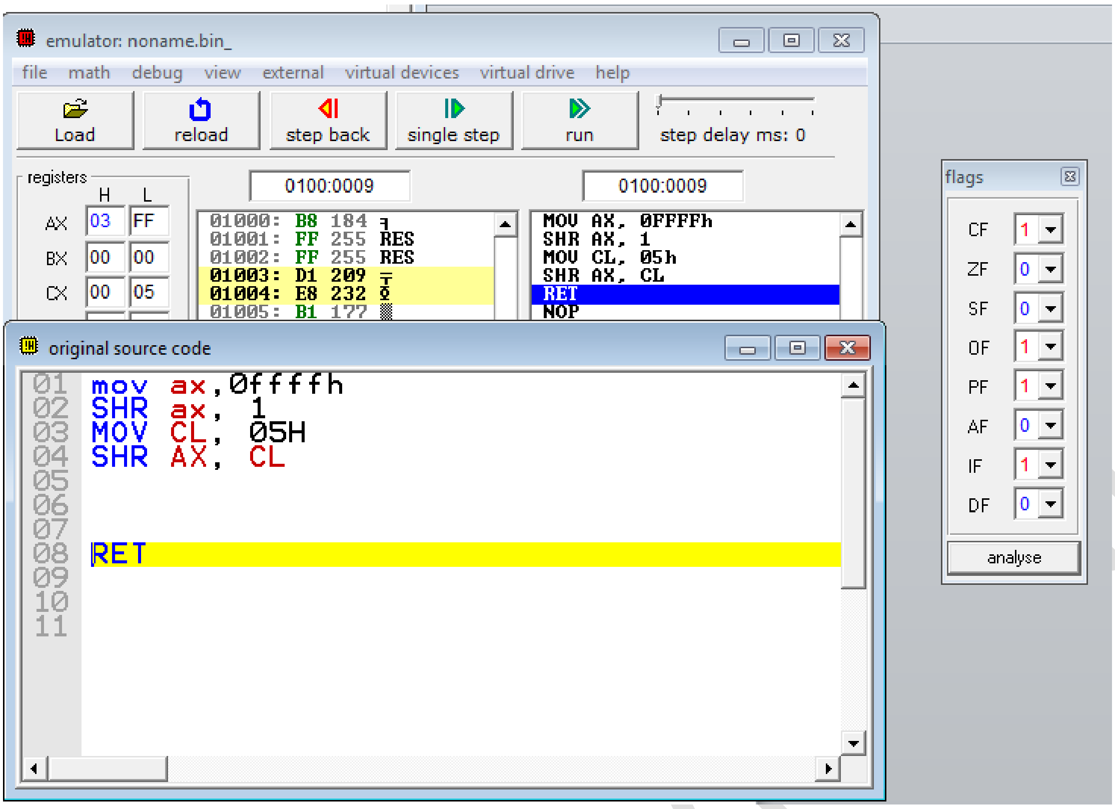
0100:0009

```
MOV  BX, 0F0F7h
SHL  BX, 1
MOV  CL, 02h
SHL  BX, CL
RET
NOP
```

flags: CF 1, ZF 0, SF 1, OF 0, PF 1, F 0, F 1, F 0

```
mov bx,0f0f7h
SAL bx, 1
mov cl,2
SAL bx, CL




RET
```

SHR	memory, immediate REG, immediate memory, CL REG, CL	Shift operand1 Right. The number of shifts is set by operand2. Algorithm: • Shift all bits right, the bit that goes off is set to CF. • Zero bit is inserted to the left-most position. Example: MOV AL, 00000111b SHR AL, 1 ; AL = 00000011b, CF=1. RET C O r r OF=0 if first operand keeps original sign.



	H	L
AX	03	FF
BX	00	00
CX	00	05

0100:0009

```
01000: B8 184 ╕
01001: FF 255 RES
01002: FF 255 RES
01003: D1 209 ╤
01004: E8 232 Φ
01005: B1 177 ▒
```

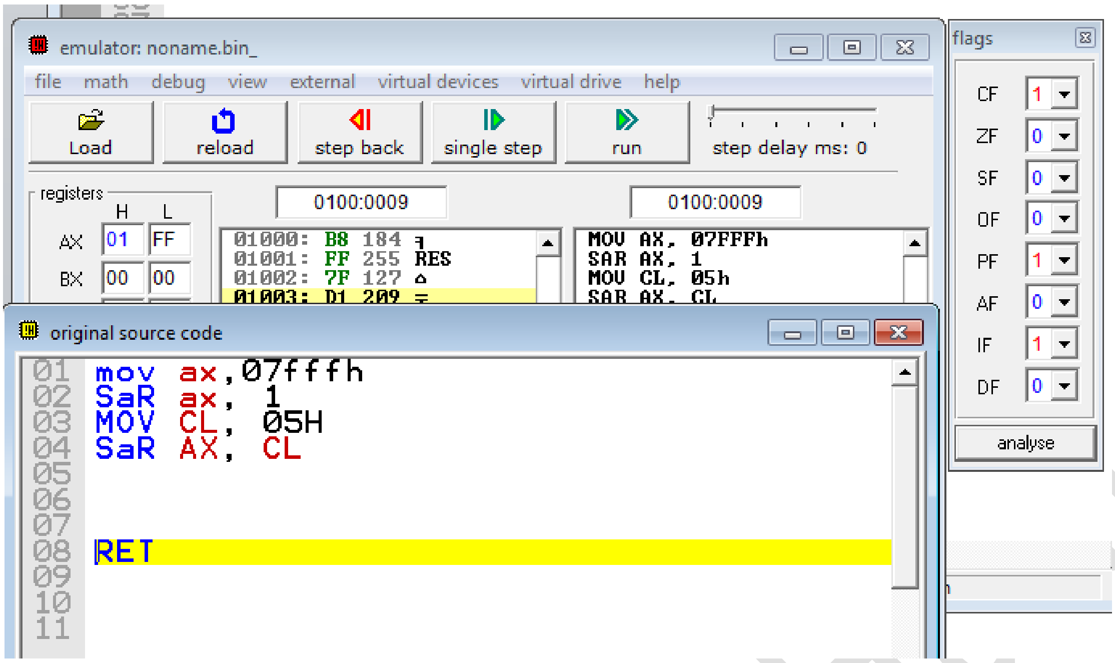
0100:0009

```
MOV  AX, 0FFFFh
SHR  AX, 1
MOV  CL, 05h
SHR  AX, CL
RET
NOP
```

```
mov ax,0ffffh
SHR ax,  1
MOV CL,  05H
SHR AX,  CL



RET
```

CF	1
ZF	0
SF	0
OF	1
PF	1
AF	0
IF	1
DF	0

SAR	memory, immediate REG, immediate memory, CL REG, CL	Shift Arithmetic operand1 Right. The number of shifts is set by operand2. Algorithm: • Shift all bits right, the bit that goes off is set to CF. • The sign bit that is inserted to the left-most position has the same value as before shift. Example: MOV AL, 0E0h ; AL = 11100000b SAR AL, 1 ; AL = 11110000b, CF=0. MOV BL, 4Ch ; BL = 01001100b SAR BL, 1 ; BL = 00100110b, CF=0. RET C O r r OF=0 if first operand keeps original sign.



registers

	H	L
AX	01	FF
BX	00	00

0100:0009

```
01000: B8 184 ╕
01001: FF 255 RES
01002: 7F 127 ⌂
01003: D1 209 ╤
```

0100:0009

```
MOV AX, 07FFFh
SAR AX, 1
MOV CL, 05h
SAR AX, CL
```

```
mov ax,07fffh
SaR ax,  1
MOV CL,  05H
SaR AX,  CL



RET
```

CF	1
ZF	0
SF	0
OF	0
PF	1
AF	0
IF	1
DF	0

ROL	memory, immediate REG, immediate memory, CL REG, CL	Rotate operand1 left. The number of rotates is set by operand2. Algorithm: shift all bits left, the bit that goes off is set to CF and the same bit is inserted to the right-most position. Example: MOV AL, 1Ch ; AL = 00011100b ROL AL, 1 ; AL = 00111000b, CF=0. RET C O r r OF=0 if first operand keeps original sign.
ROR	memory, immediate REG, immediate memory, CL REG, CL	Rotate operand1 right. The number of rotates is set by operand2. Algorithm: shift all bits right, the bit that goes off is set to CF and the same bit is inserted to the left-most position. Example: MOV AL, 1Ch ; AL = 00011100b ROR AL, 1 ; AL = 00001110b, CF=0. RET C O r r OF=0 if first operand keeps original sign.

RCL	memory, immediate REG, immediate memory, CL REG, CL	Rotate operand1 left through Carry Flag. The number of rotates is set by operand2. When **immediate** is greater then 1, assembler generates several **RCL xx, 1** instructions because 8086 has machine code only for this instruction (the same principle works for all other shift/rotate instructions). Algorithm: shift all bits left, the bit that goes off is set to CF and previous value of CF is inserted to the right-most position. Example: STC ; set carry (CF=1). MOV AL, 1Ch ; AL = 00011100b RCL AL, 1 ; AL = 00111001b, CF=0. RET C O r r OF=0 if first operand keeps original sign.
RCR	memory, immediate REG, immediate memory, CL REG, CL	Rotate operand1 right through Carry Flag. The number of rotates is set by operand2. Algorithm: shift all bits right, the bit that goes off is set to CF and previous value of CF is inserted to the left-most position. Example: STC ; set carry (CF=1). MOV AL, 1Ch ; AL = 00011100b RCR AL, 1 ; AL = 10001110b, CF=0. RET C O r r OF=0 if first operand keeps original sign.
ROL	memory, immediate REG, immediate memory, CL REG, CL	Rotate operand1 left. The number of rotates is set by operand2. Algorithm: shift all bits left, the bit that goes off is set to CF and the same bit is inserted to the right-most position. Example: MOV AL, 1Ch ; AL = 00011100b

		ROL AL, 1 ; AL = 00111000b, CF=0. RET C O r r OF=0 if first operand keeps original sign.
ROR	memory, immediate REG, immediate memory, CL REG, CL	Rotate operand1 right. The number of rotates is set by operand2. Algorithm: shift all bits right, the bit that goes off is set to CF and the same bit is inserted to the left-most position. Example: MOV AL, 1Ch ; AL = 00011100b ROR AL, 1 ; AL = 00001110b, CF=0. RET C O r r OF=0 if first operand keeps original sign.