**Practical - 6**

**Aim:**

Write an assembly program to perform following operations on variable A and save each result in separate variable. Also analyze the results.

1. Logical left shift
2. Logical right shift
3. Arithmetic left shift
4. Arithmetic right shift
5. Rorate left with carry
6. Rorate left without carry
7. Rorate right with carry
8. Rorate right without carry

**Description of instructions used:**

* **Logical left shift:**

; logical left shift

; 9876 --> 1001 1000 0111 0110

; 0011 0000 1110 1100 --> 30EC in first shift

; 0110 0001 1101 1100 --> 61DC in second shift

* **Logical right shift:**

; logical right shift

; 9876 --> 1001 1000 0111 0110

; 0100 1100 0011 1011 --> 4C3B

* **Arithmetic left shift:**

; arithmetic left shift

; 9876 --> 1001 1000 0111 0110

; 0011 0000 1110 1100 --> 30EC

* **Arithmetic right shift:**

; arithmetic right shift

; 9876 --> 1001 1000 0111 0110

; 1100 1100 0011 1011 --> CC3B

Instructions to perform shift operations

* **SHL/SAL** − Used to shift bits of a byte/word towards left and put zero(S) in LSBs.
* **SHR** − Used to shift bits of a byte/word towards the right and put zero(S) in MSBs.
* **SAR** − Used to shift bits of a byte/word towards the right and copy the old MSB into the new MSB.

Instructions to perform rotate operations

* **ROL** − Used to rotate bits of byte/word towards the left, i.e. MSB to LSB and to Carry Flag [CF].
* **ROR** − Used to rotate bits of byte/word towards the right, i.e. LSB to MSB and to Carry Flag [CF].
* **RCR** − Used to rotate bits of byte/word towards the right, i.e. LSB to CF and CF to MSB.
* **RCL** − Used to rotate bits of byte/word towards the left, i.e. MSB to CF and CF to LSB.

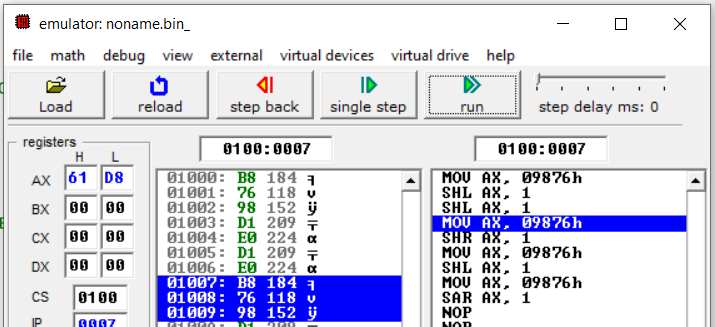
**Code:**

1. **Logical left shift:**

MOV AX,9876h

SHL AX,2

**Output:**

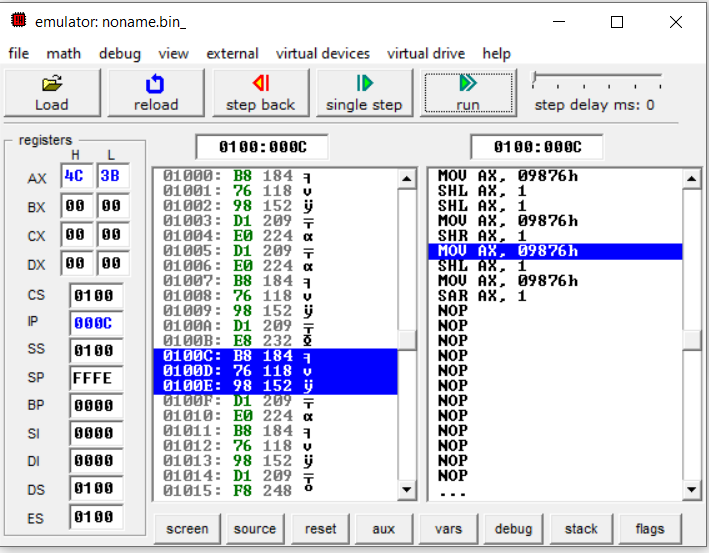


1. **Logical right shift :**

MOV AX,9876h

SHR AX,1

**Output:**

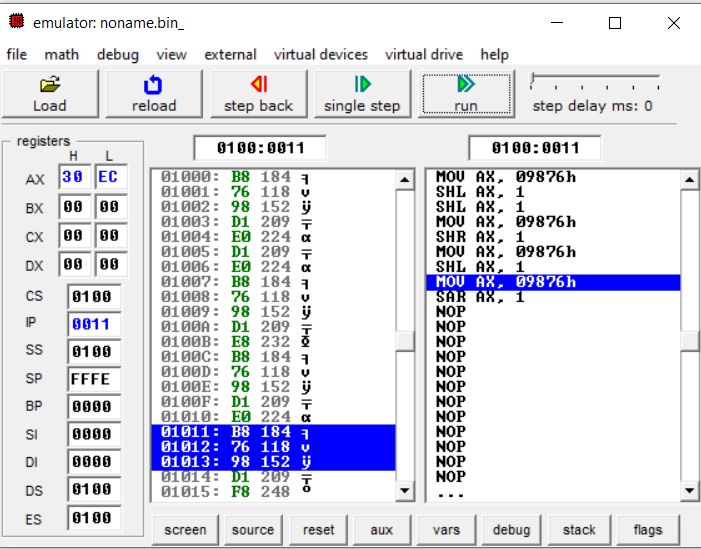


1. **Arithmetic left shift**

MOV AX,9876h

SAL AX,1

**Output:**

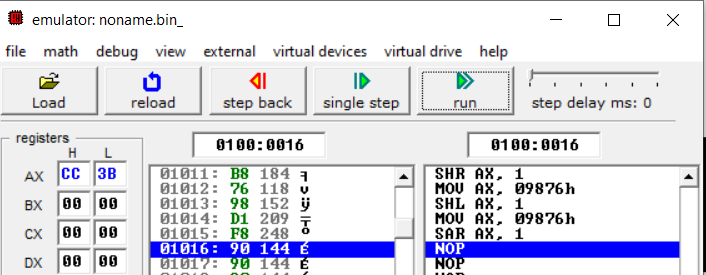


1. **Arithmetic right shift :**

MOV AX,9876h

SAR AX,1

**Output:**

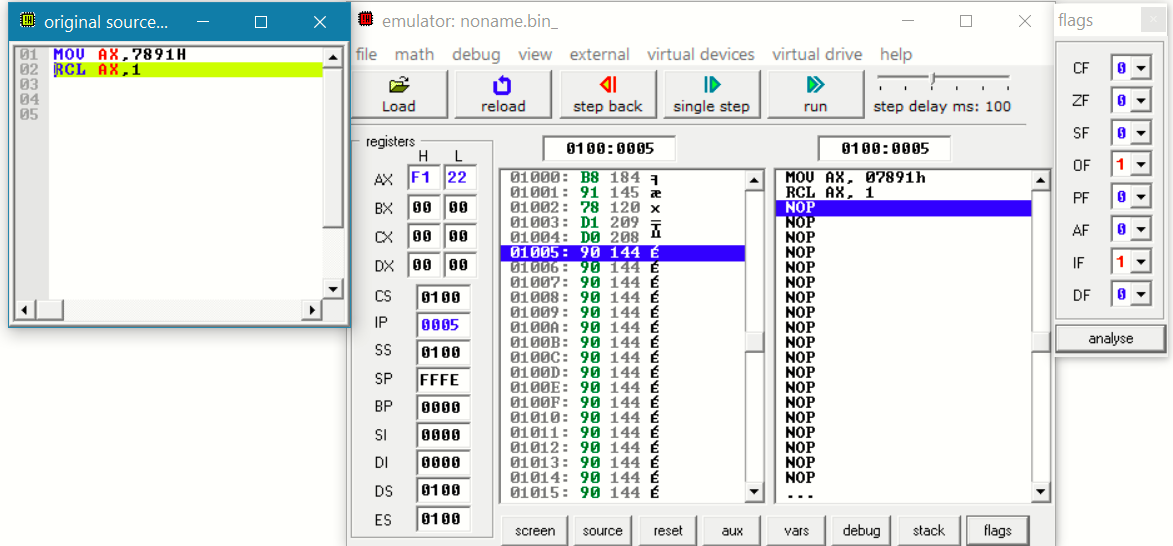


1. **Rotate left with carry:**

MOV AX,7891H

RCL AX,1

**Output:**

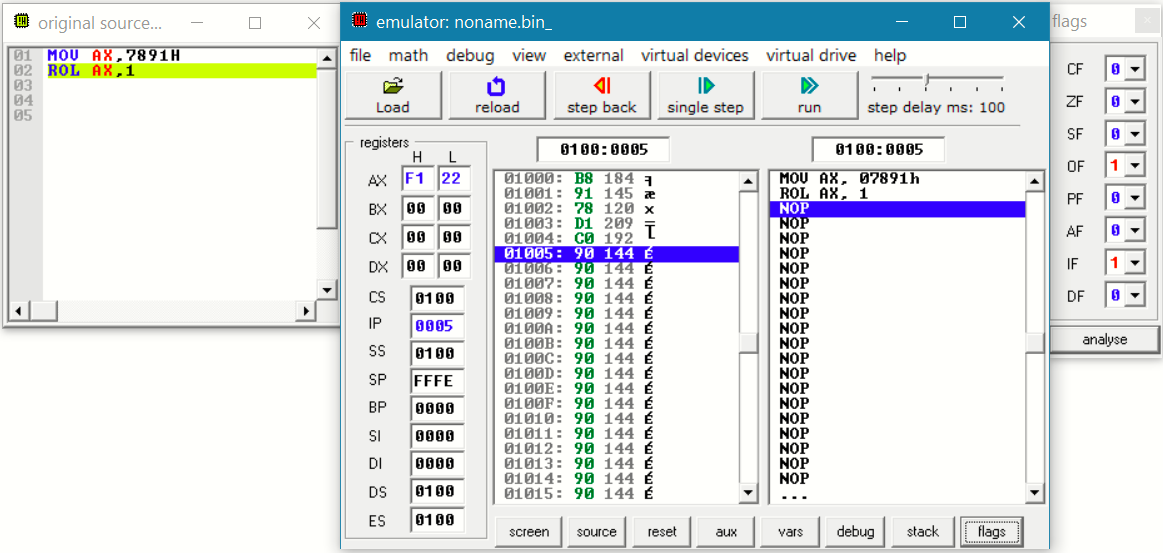


1. **Rorate left without carry:**

MOV AX,7891H

ROL AX,1

**Output:**

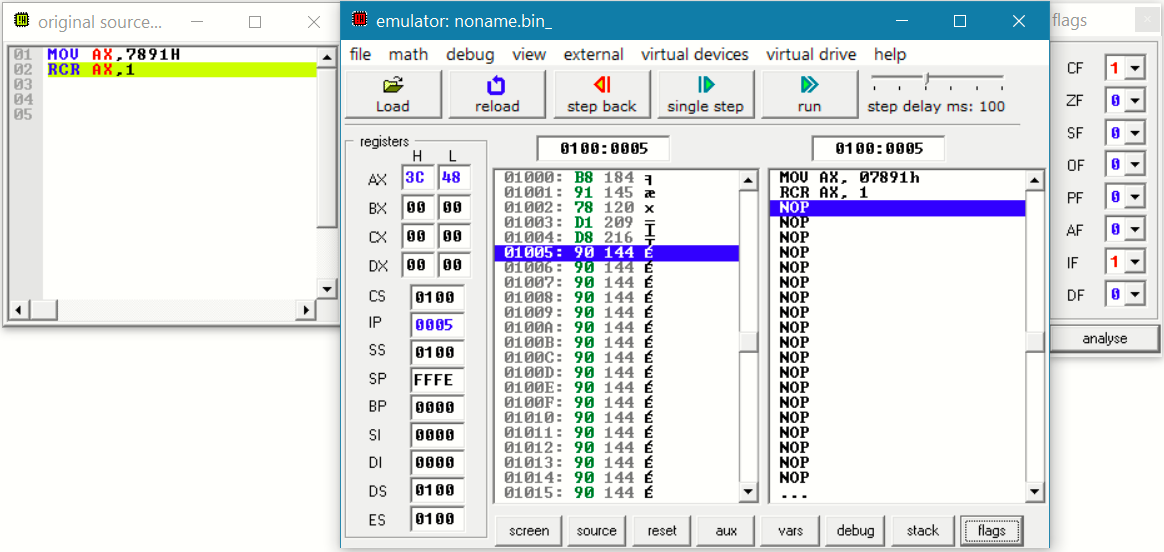


1. **Rorate right with carry:**

MOV AX,7891H

RCR AX,1

**Output:**



1. **Rorate right without carry:**

MOV AX,7891H

ROR AX,1

**Output:**

