

# Homework 8

10/24/23  
AIDAN KISER

1. 1 mov cl, 1 ; CL = 1
- 2 mov al, 1Eh ; AL = 0x1E
- 3 rol al, cl ; AL = 0x3E
- 4 shl al, cl ; AL = 0x28
- 5 mov al, 0A4h ; AL = 0xA4
- 6 mov cl, 2 ; CL = 2
- 7 ror al, cl ; AL = 0x91
- 8 shr al, cl ; AL = 0x24
- 9 stc
- 10 mov al, 86h ; AL = 0x86
- 11 mov cl, 1 ; CL = 1
- 12 rcl al, cl ; AL = 0xOD
- 13 stc
- 14 mov al, 36h ; AL = 0x36
- 15 mov cl, 1 ; CL = 1
- 16 rcr al, cl ; AL = 0x1B

## Instructions:

1. Sets CL to 1
2. Sets AL to 0x1E
3. Rotates the bits in AL to the left by CL positions.

Since CL is 1, the LSB is moved to the MSB.

4. Shifts the bits in AL to the left by CL positions.

Since CL is 1, the left shift is 1.

5. Sets AL to 0xA4.
6. Sets CL to 2.

7. Rotates the bits in AL to the right by CL positions.

Since CL is 2, AL is shifted to the right by 2 positions.

8. Shifts the bits in AL to the right by CL positions.

CL is 2, this is equivalent to a single right shift.

10. AL is set to 86h.

11. CL is set to 1.

12. Rotates the bits in AL to the left by CL positions.

CL is 1, so AL is rotated to the left by 1 position.

14. AL is set to 36h.

15. CL is set to 1.

16. Rotates the bits in AL to the right by CL positions.

CL is 1, so AL is rotated to the right by 1 position.

3 a)

mov dx, 0	; dx = 0
mov ax, 80A0h	; ax = 0x80A0
mov cx, 3h	; cx = 0x3
mul cx	; dx:ax

The "mul" instruction multiplies AX by CX and stores the result in AX and the high part of the result in DX.

\* If DX is not set to 0 before the multiplication, the contents of DX will affect the final result.

$$AX = 0 \times 80A0 \times 0 + 3 = 0 \times 24280$$

$$\therefore DX = \text{High 16 bits} = 0 \times 02$$

$$AX = \text{Low 16 bits} = 0 \times 4280$$

b) "IDIV" is used for signed integer division. It can cause an overflow if the result of the division cannot be represented within the range of a signed number. An overflow occurs when the quotient or remainder is too large or too small to fit within the destination operand size.

c) `mov ax, 0h ; AX = 0x0`

`mov dx, 0h ; DX = 0x0`

`sub ax, 3h ; subtract 3 from AX. AX = 0xFFFFD`

`sbb dx, 0 ; subtract the carry flag from AX.`

• DX is still 0. But AX becomes 0xFFFFD from 0-0.

• This sequence of instructions can be used for error checking, arithmetic operations, mode addressing for borrowing from registers, and handling signed integer operations where the carry flag indicates whether a borrow occurred during subtraction.

4. This .asm program implements a Case Table, which is essentially a table that associates a set of characters with corresponding procedures. The Case Table is used to map input characters A, B, C, D to specific procedures, allowing it to

execute the appropriate code based on user input.

The user is prompted to enter a character and then calls the corresponding while displaying a message associated with the chosen character.

- \* • The disadvantage of manually putting a value for "Entry Size" and "Number Of Entries" is that it can lead to errors and maintenance difficulties.  
The approach taken in the program ensures the values will be correct and you won't have to keep track of the values yourself.