AND   op1, op2

OR

XOR

NOT

SHIFT RIGHT   SHR   op, ? $\lessgtr$ 3
                                    CL

SHIFT LEFT   SHL   op, ?

ROTATE RIGHT ROR   op, ?

ROTATE LEFT   ROL   op, ?

1010 0111
          2
              CF

              CF
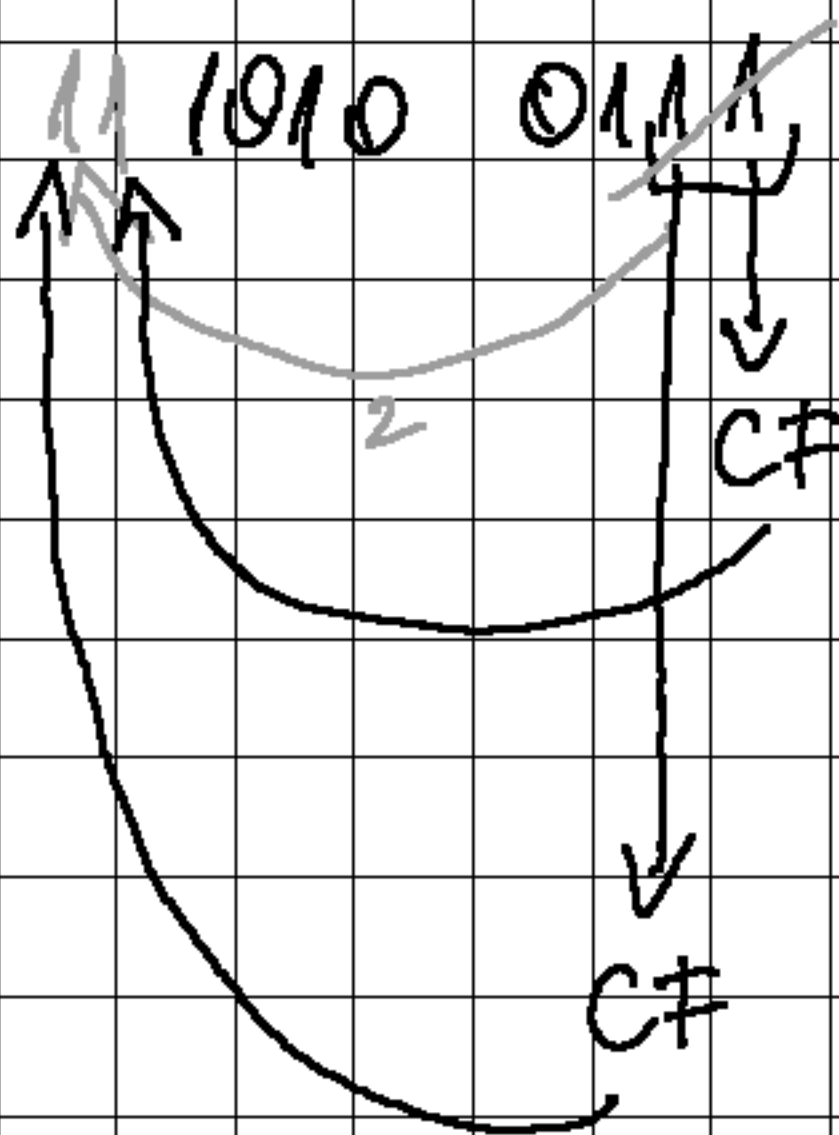
ROTATE WITH CF RIGHT   RCR

ROTATE WITH CF LEFT   RCL

X↑ ↗↗ 11 00   11 01
        ↓↓↓ CF
                ↓ CF
            ↓ CF
        ↓
        CF
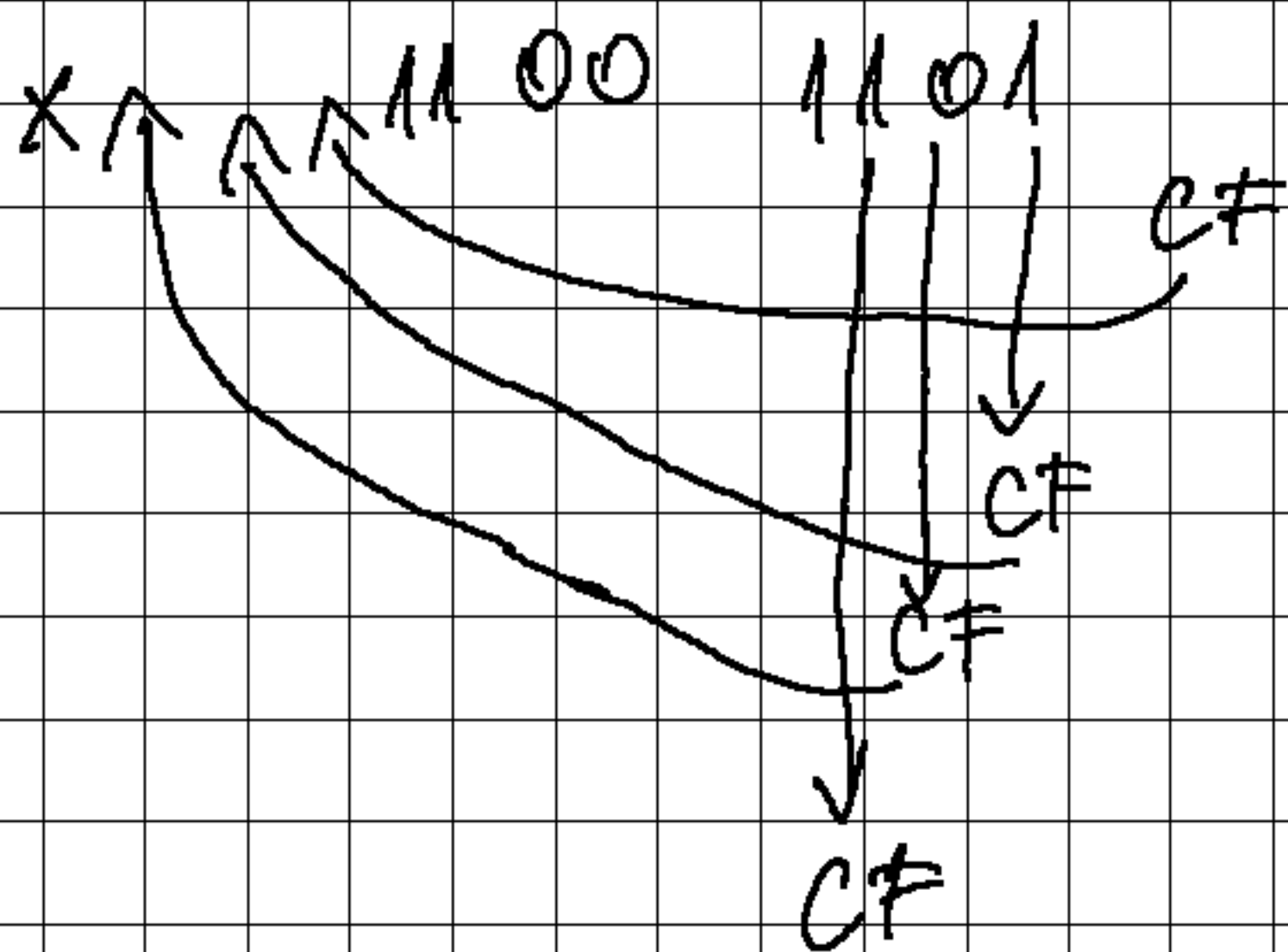
mov al ,   0011 0011 b
mov cl , 2
shl al , cl

⇒ 33 h

AX | 0ABCD h
        ↓
    1010   1011   1100   1101

    15 ----------- 7654   32 10
                          ↓↓↓

AND AX , 0000 0000 0000 0111 → isolate
isolate   for pos 0-2
for 7-12 position :   by making
                                     a mask
AND AX, 0001 1111 1000 0000

isolate

for 7-12 position: then move them to pos 1-6

AND AX, 0001 1111 1000 0000

→6
SHIFT

SHR AX, 6

Ex:

    mov AX, [0i]

    mov BX, [0i]

    mov DX, 0

0-2 DX ⟺ 10-12 A

bitii 0-2 din Dx corespund cu 10-12 din A

AND AX, 0001 11 0000000000

SHR AX, 10   12 10

OR DX, AX

biți din DX 3-6 have the val 1.

OR DX   0000,0000,0111,1000

biți 7-10 corespund cu biți 1-4 din [a]

    Mov AX, [a]

    AND AX, 0000,0000,0001,1110

    SHL AX,6

    OR DX, AX

biți 11-12 au val 0

    AND DX, 1110011111111111b

---

    mov ax, 86h $\overset{=}{}$ 0086h

    cbw

    mov bl, 2

    div bl

                        with cbw

    0000 0000 1000 0110b

cbw ⇒ 1111 1111 1000 0110b
         7   F   86 /:2 ⇒ 7893h
                              in
                              AL

              (in AL max 2 digits in hexa)

unsigned + mov ax, 0
           MUL , DIV

signed ⎰ +   cbw, cwd, cwde, cdq
       ⎱ −   iDIV , iMUL

a, b, c — byte
                        a*b+ c
e, f, g — word

mov al, [a]   ⇒ 90h

mul byte [b]   ⇒ 2h ⇒ AX: 180h ⇒ crashes

~~add al, [c]~~

~~mov DX , [a]~~

mov bl, [c]

mov bh, 0

⎰ add bx
⎱ mov bx, 0
  mov bl, [c]

a, b, c — byte

l, f, g — word

mov al, -3 ⇒ 1111 1101

mov bl, 2

mul bl ⇒ FC * 2 ⇒ AX