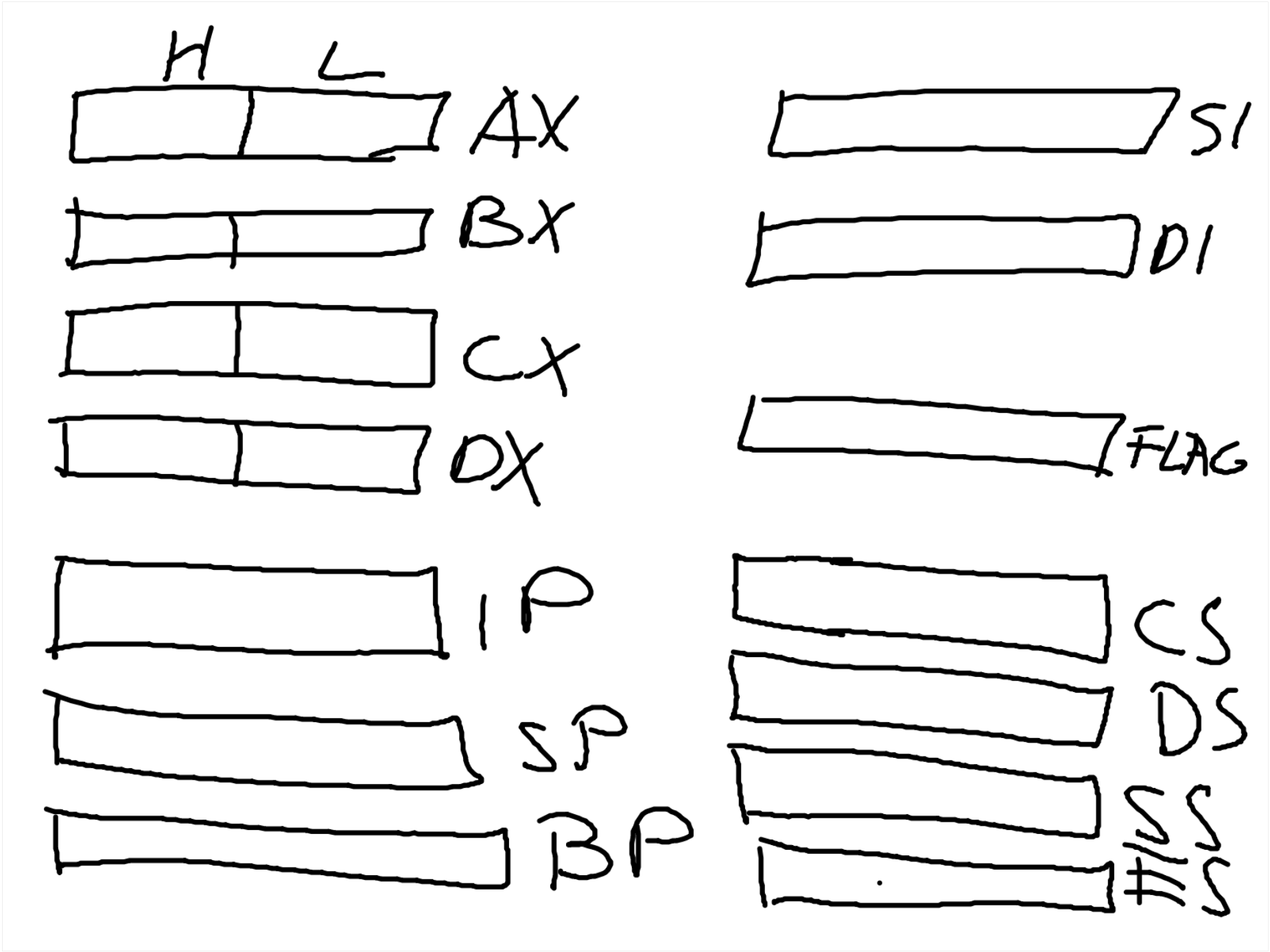
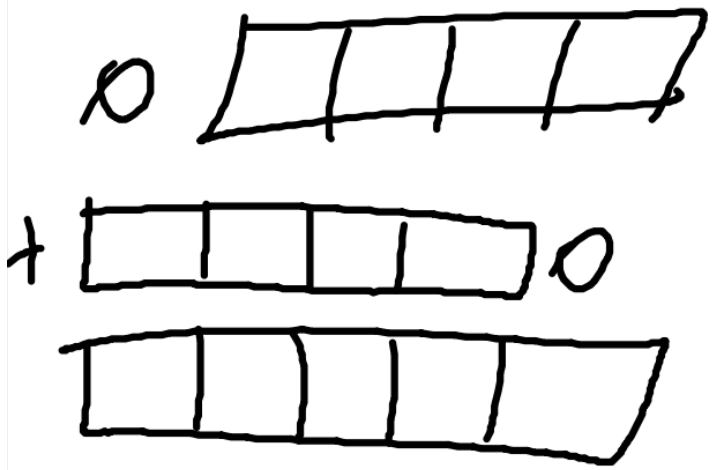

H
L
AX
BX
CX
DX
SI
DI
FLAG
IP
SP
BP
CS
DS
SS
ES



ADATMOZGATÓ

MOV AL, BL ;

MOV DH, CL

MOV BL, 7

00000111B

-8

23-12+3

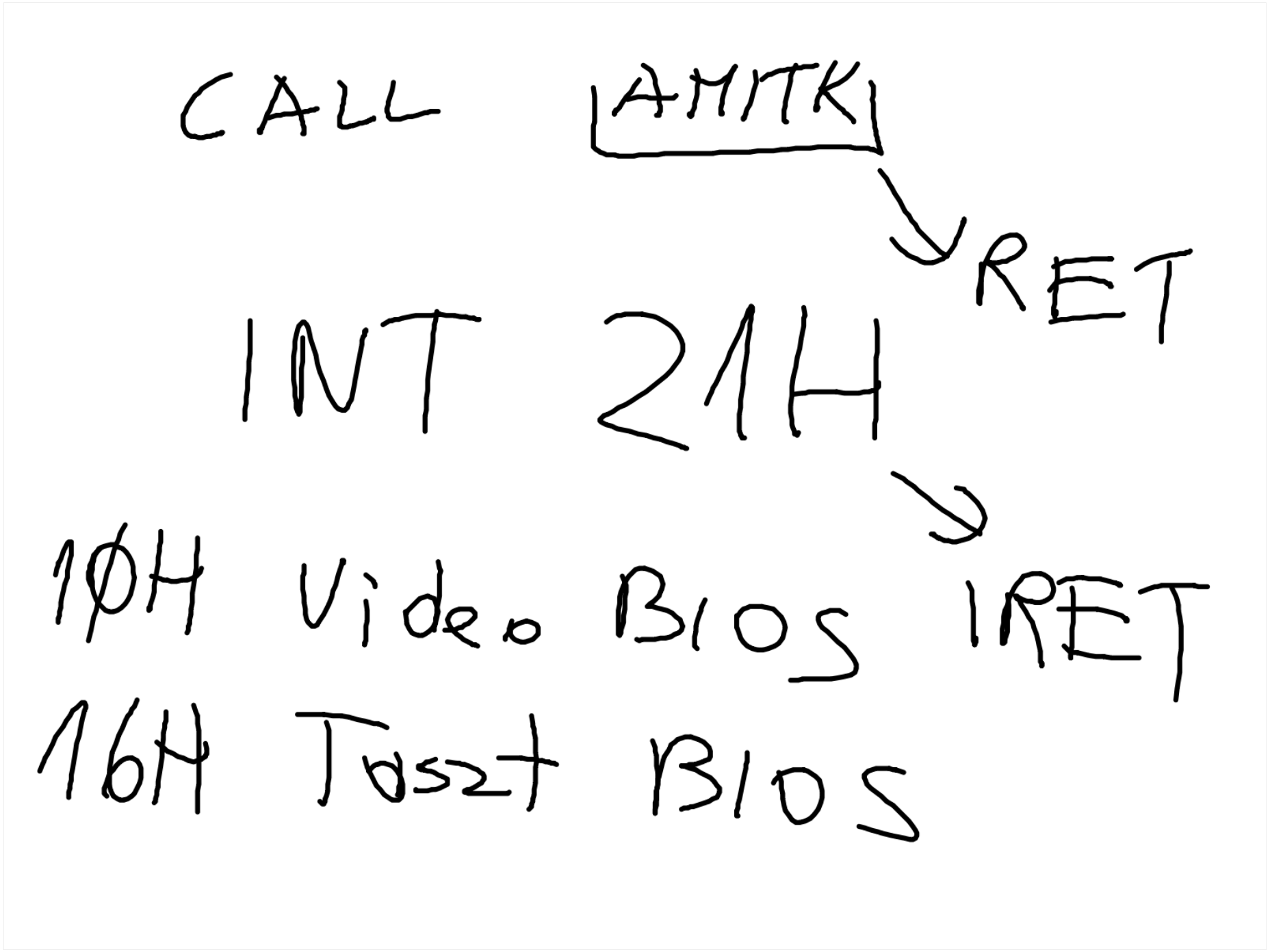
LAC 1

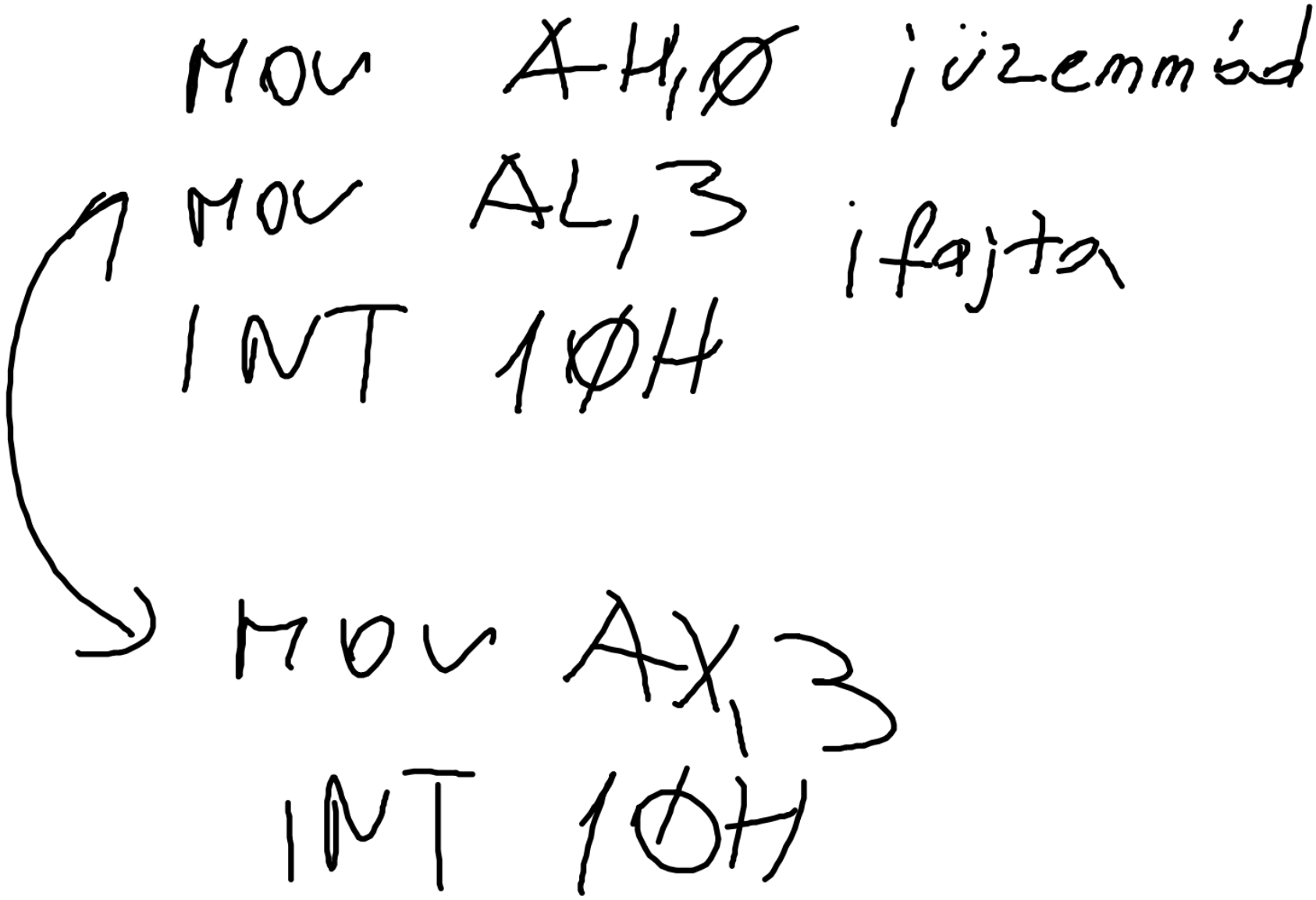
MOV DX, 0BABAH

MOV AL, SS:[BX]

```
MOV  AL, DH
MOV  DH, AL
~~MOV  5, AL~~
~~MOV  DH, 51~~
MOV  DS, AX
~~MOV  DS, 2~~
```

```
push   DX
pop    WORD PTR [BX]
pushf
popf
XCHG   AX,SI
```


CALL
AMITK
RET
INT 21H
IRET
10H Video BIOS
16H Teszt BIOS


MOV AH,0 ;üzemmód
MOV AL,3 ;fajta
INT 10H
MOV AX,3
INT 10H

```
mov  AL, 'S'
mov  DX, 'AB'
```

Direktiva: ORG, END

DB

```
TT DB 'B', 7, -1
```

```
DB 'Szervusz!'
DB 13,10
DB 37 DUP('C')
DB 200 DUP(?)
```

DN 5,65000,-31000

DD 7,77,3.2,0.

DQ .13,7.7,8

DT 2.13

~~MOV BX, ITT~~

MOV BX, OFFSET ITT

String kiiRatds

MOV AH, 9
MOV DX, OFFSET ....
INT 21H

LABEL DIREKTIVA

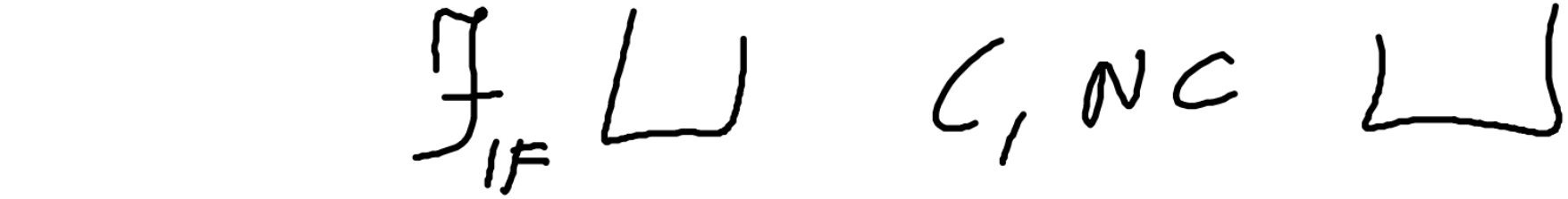
IP-t módosítja

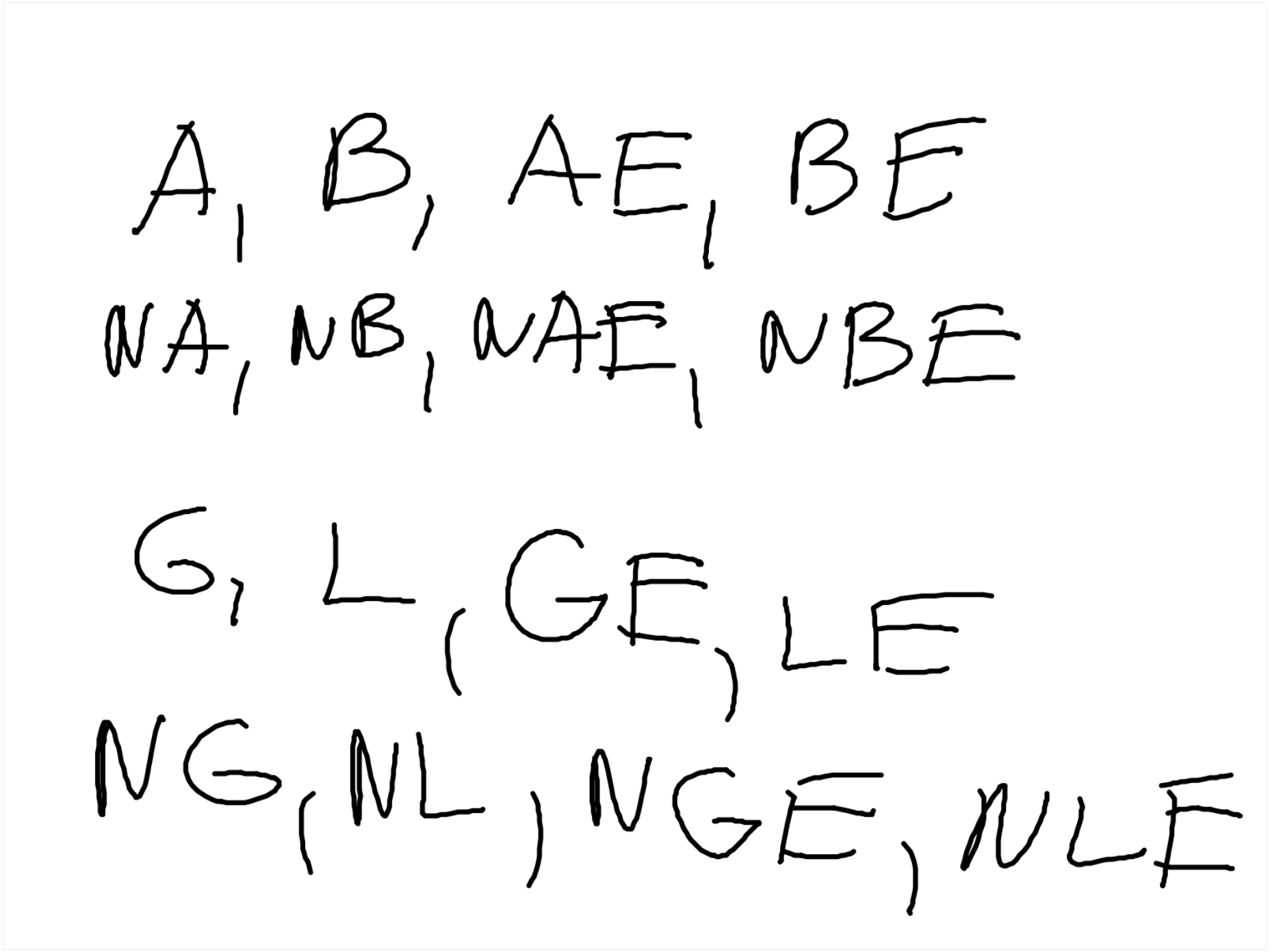
INT ; IRET

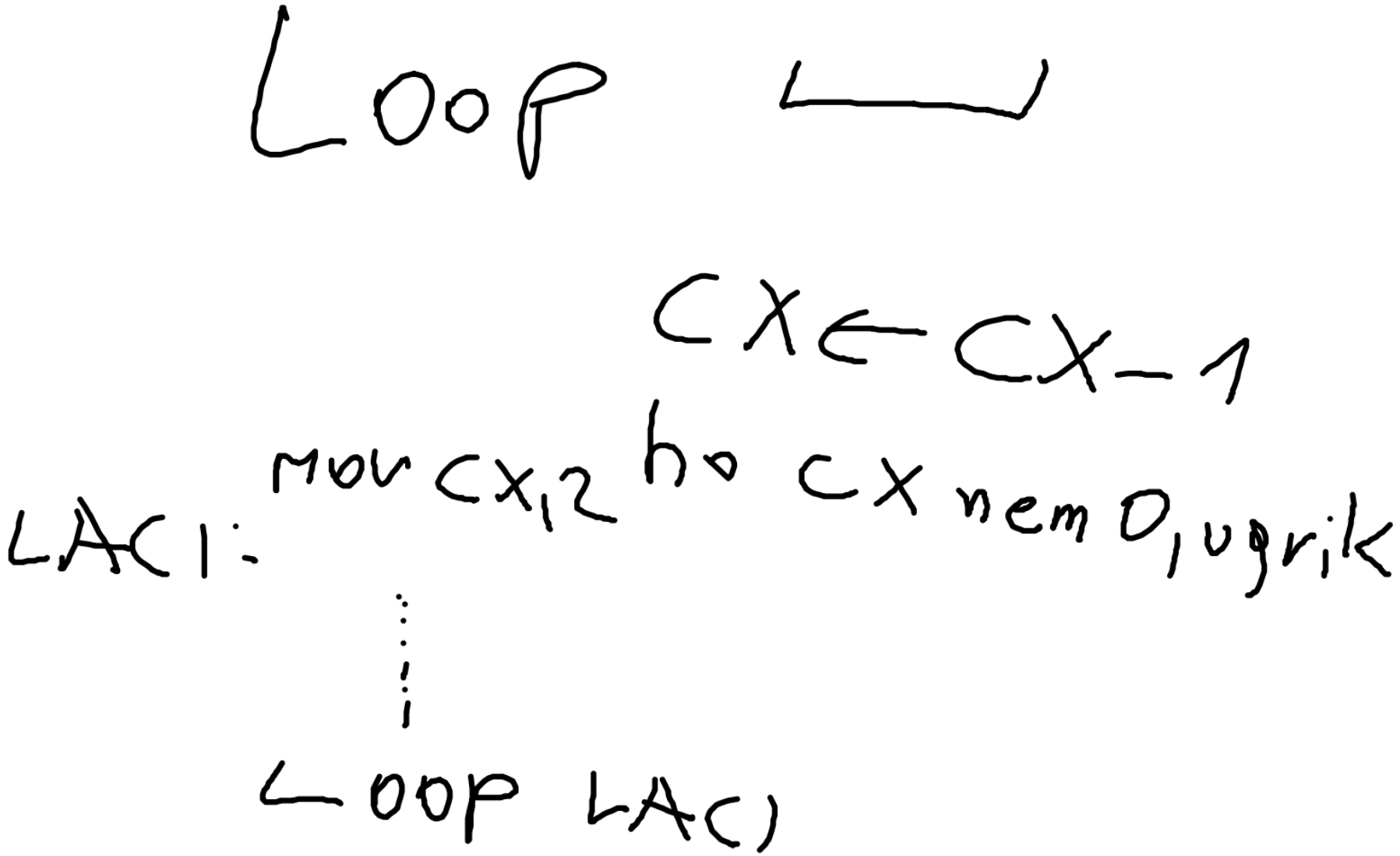
JMP ┌──┐

CALL ┌──┐

RET ; RETF


IF
C, NC
NM, NP
Z, NZ
S, NS
PO, NPO
O, NO
PE, NPE
M, P


A, B, AE, BE
NA, NB, NAE, NBE
G, L, GE, LE
NG, NL, NGE, NLE


Loop
CX ← CX - 1
mov cx,12 ho cx nemo,ugrik
LAC1:
LOOP LAC1

LOOPE v. LOOPZ

LOOPNE v LOOPNZ

JZ MESSZE

MOV AX,5

JCXZ

|

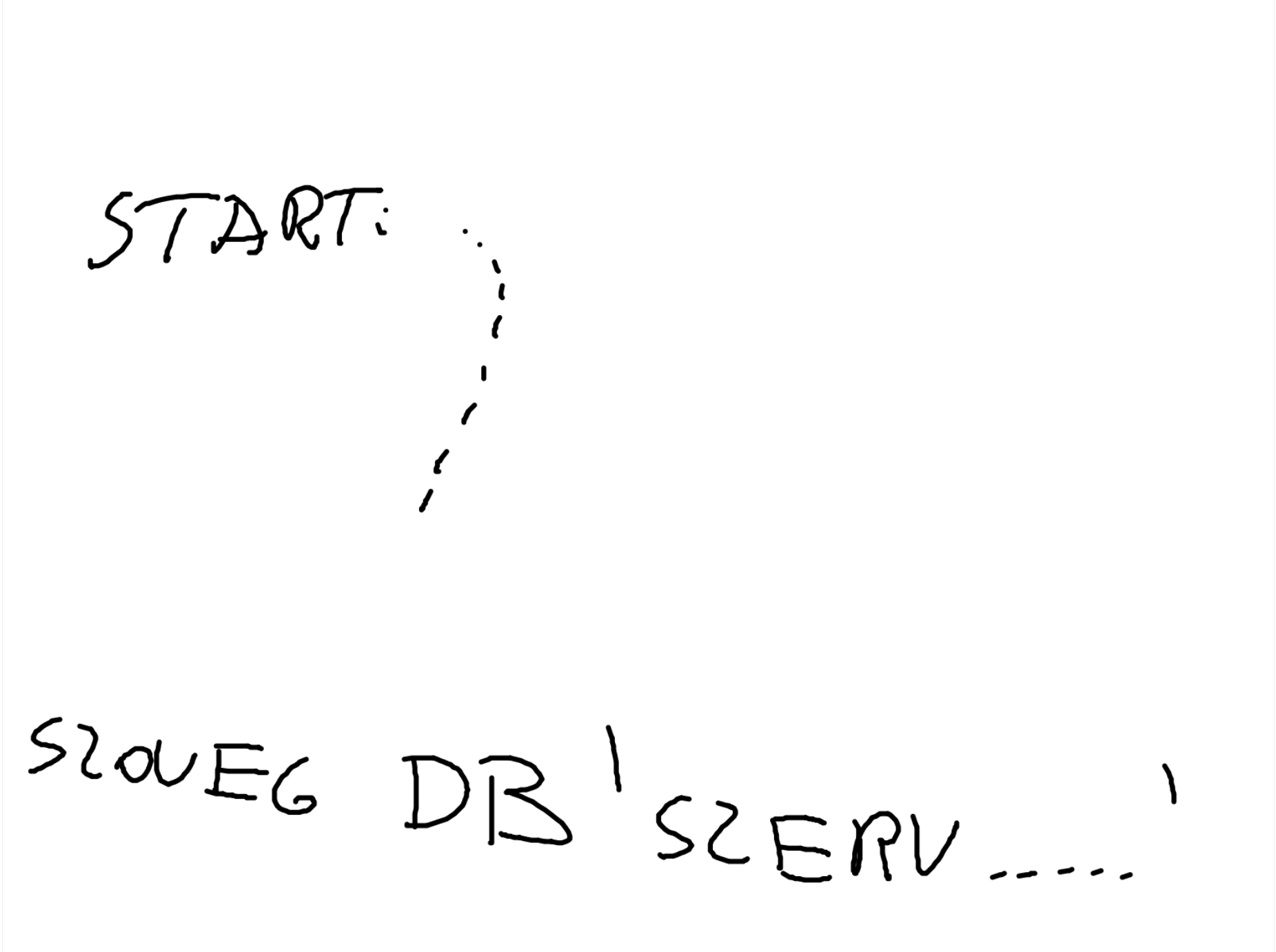
JNZ AT

JMP MESSZE

AT:

MOV AX,5

```
ORG 100H

START:

        - - - ,
        . - - ;
        - - - .

BUBU:   ; BUBU LABEL CODE
```


START:
SZOVEG DB 'SZERV.....'

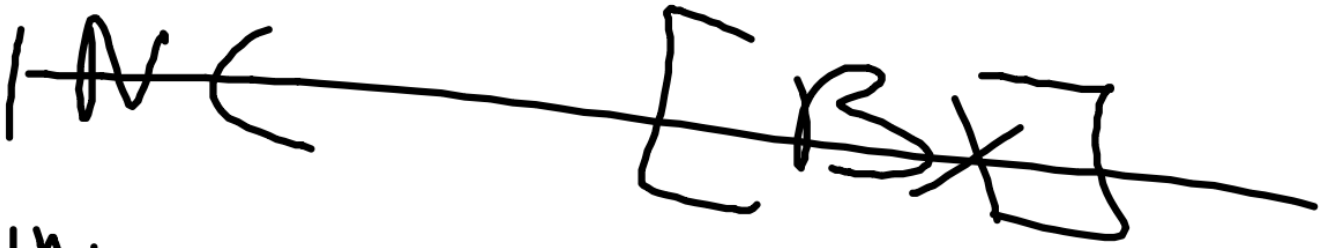
MENNY1 EQU 10

MOV CX, MENNY1

ARITMETIKA I

INC

DEC



INC BYTE PTR [BX]

INC WORD PTR [BX]

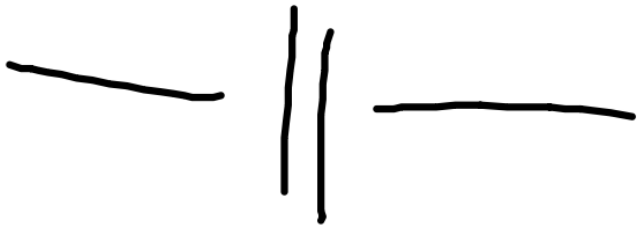
NEG

ADD cdl, forrds

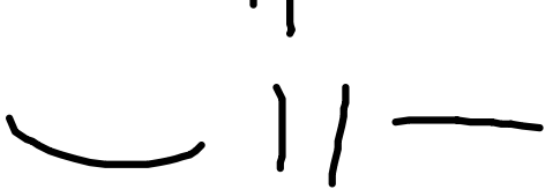
SUB cdl, forrds

ADC

SBB



CMP



IMUL, MUL 8/16

AX ← AL * 8bit

DX:AX ← AX * 16bit

IDIV, DIV

8/16

AL ← HANY.

AH ← mar.

AX : 8bites

AX ← HANY.

DX ← MAR.

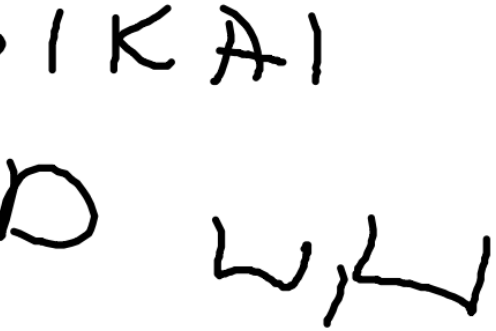
DX:AX : 16bit

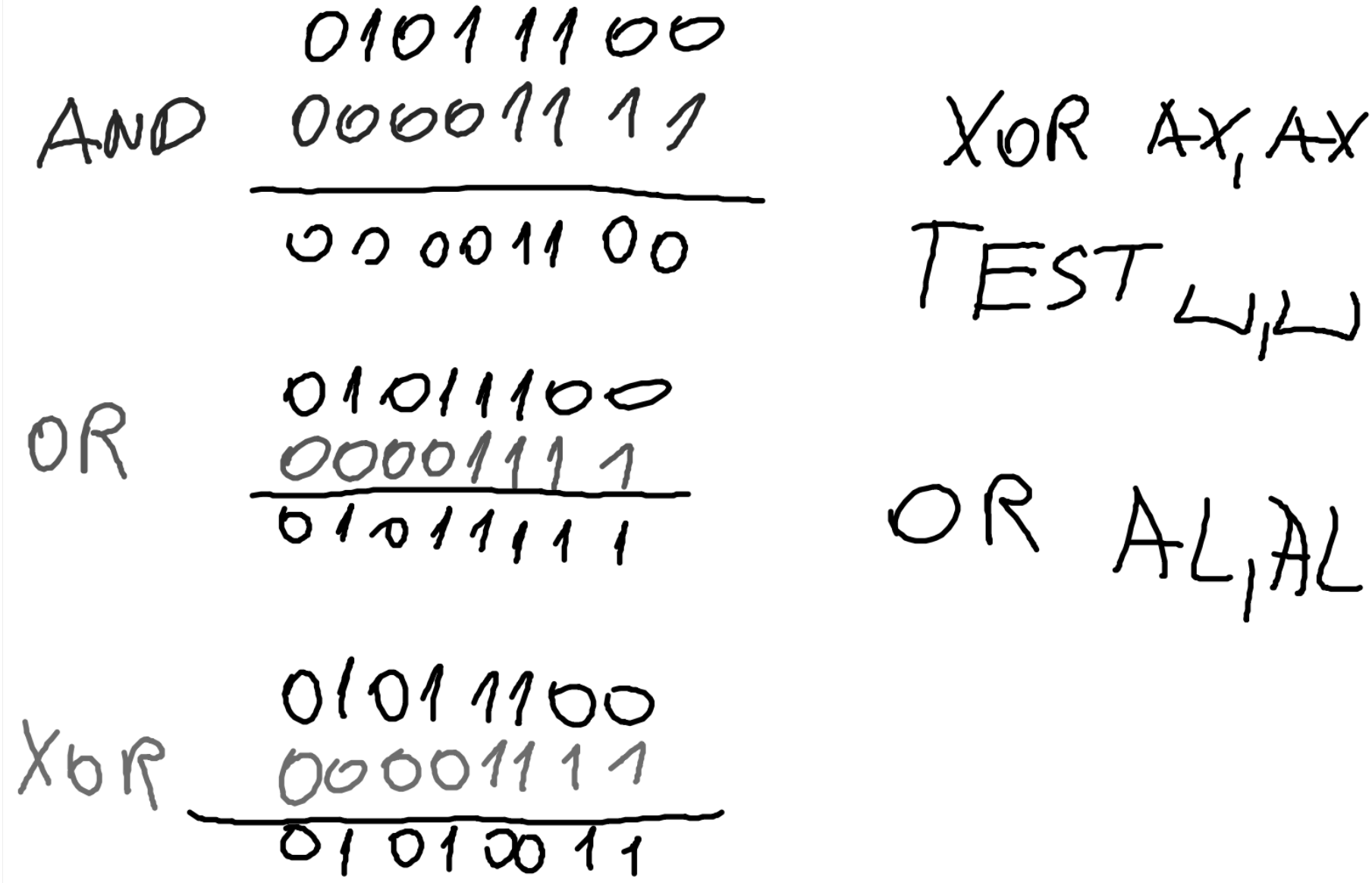
CMP AL,'Q'

JZ KILEP

LOGIKAI MŰVELETEK

AND




01011100
AND 00001111
00001100
XOR AX, AX
TEST AL, AL
01011100
OR 00001111
01011111
OR AL, AL
01011100
XOR 00001111
01010011

```
NOT ↵      ; XOR ↵,0FFH
 |||
XOR ↵,-1

MOV DL,'↵'
MOV AH,2
INT 21H
```

```
MOV  AH,2 ; karakterkiirds

MOV  DL,BL
INT  21H
MOV  DL,'+'
INT  21H
MOV  DL,BH
INT  21H
MOV  DL,'='
INT  21H
```

ASCII elfüntetes

```
AND  BL, 00001111B
XOR  BL, '0'
SUB  BL, '0'
```

0011 XXXX → '0' .. '9'

0011 0000 → '0'

```
AND  BX, 0000111100001111B
```

ADD BL, BH

Karakteresítés

ADD BL, '0'

XOR BL, '0'

OR BL, '0'

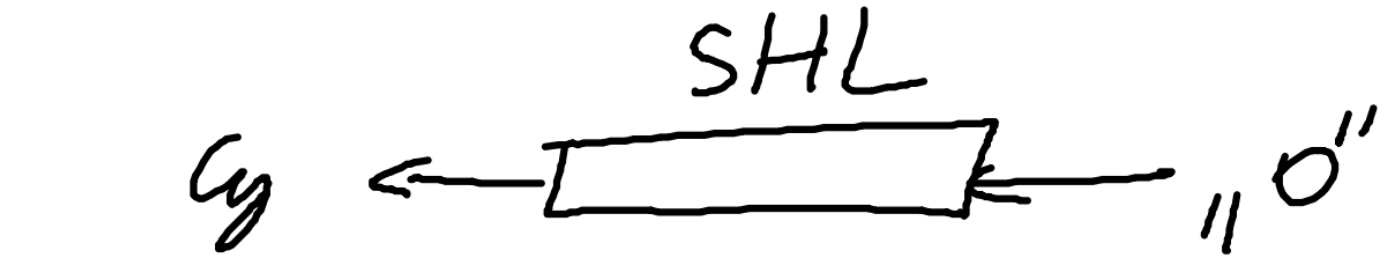
00001000

00110000

```
MOV DL, BL
INT 21H
```

```
AND BL, 00001111B
ADD BL, BH
```

Léptető és forgató

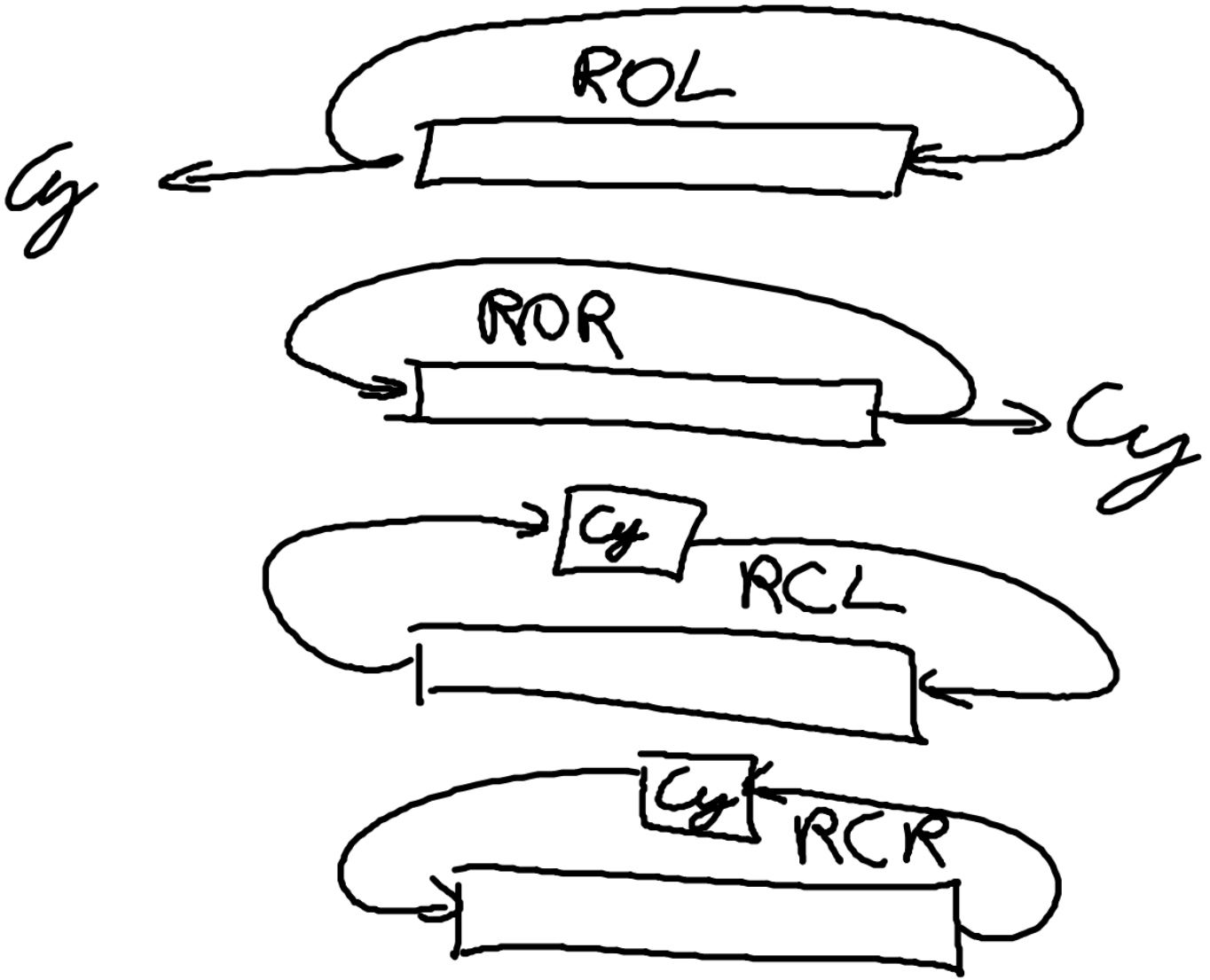


SHL CX,1

SHR BP,1

SHL, AX, CL

SAL ≡ SHL


ROL
Cy
ROR
Cy
Cy
RCL
Cy
RCR

HEXA

12345678 MOV BX, OFFSET B52

MOV AX, [BX]

ADD [BX], AX

ACF0

MOV AX, [BX+2]

2468 ADC [BX+2], AX

SHL DWORD PTR [BX], 1

CY 01011001111100000 [BX]

1 5 9 E 0

CY ← 00100100011010001 [BX+2]

4 8 D 1

RCL DWORD PTR [BX+2], 1

EGYÉB

CLC

STC

CMC

CLI

STI

NOP

HLT

CLD

STD

```
        MOV   SI, OFFSET FOR
        MOV   DI, OFFSET CEL
        MOV   CX, MENNY1
MAC1:
        MOV   AL, [SI]
        MOV   [DI], AL
        INC   SI
        INC   DI
        LOOP  MAC1
```

String kezelő utasítások:

MOVSB ; } DS:[SI] → ES:[DI]
MOVSW ; }

REP ; i is mettles prefix

LODSB ; AL, DS:[SI]
LODSW ; AX, DS:[SI]

```
STOSB   ; ES:[DI], AL
STOSW   ; ES:[DI], AX

MOV   DI, OFFSET TAR
MOV   CX, 250
XOR   AX, AX
REP   STOSW
```

CMPSB ; DS:[SI]-ES:[DI] bytes

CMPSW ; — " — palabras

REPZ y REPE

REPNZ y REPNE

```
SCASB   ; CMP AL, ES:[DI]
SCASW   ; CMP AX, ES:[DI]
```

00..7FH → rendszer

80..FF → parancssor

```
MOV SI,80H
LODSB
CMP AL,0
JZ URES
MOV AH,0
MOV CX,AX
```

```
        MOV   SI,ROIY
        LODSB
        OR    AL,AL
        JZ    URES
        INC   SI
BEC:
        LODSB
        CMP   AL,0DH
        JZ    VEGE
```

```
MOV   SI, BOLY
LODSB
OR    AL, AL
JZ    URES
INC   SI
LODSB
CMP   AL, '0'
JB    HIBA
CMP   AL, '9'
JA    HIBA
MOV   BL, AL
```

```
LODSB
CMP  AL,'L'
JNZ  HIBA
LODSB
CMP  AL,'O'
JB   HIBA
CMP  AL,'g'
JA   HIBA
MOV  BH,AL
LOSDSB
CMP  AL,0DH
JNZ  HIBA
```

```
MOV AH, 2
MOV DL, BL
INT 21H
MOV DL, 4'
INT 21H
MOV DL, BH
INT 21H
MOV DL, 1='
INT 21H
```

```
SUB  BL, '0'
MOV  DL, BL
ADD  DL, BH
INT  21H
```

AAM

AH ← AL DIV 10

AL ← AL MOD 10

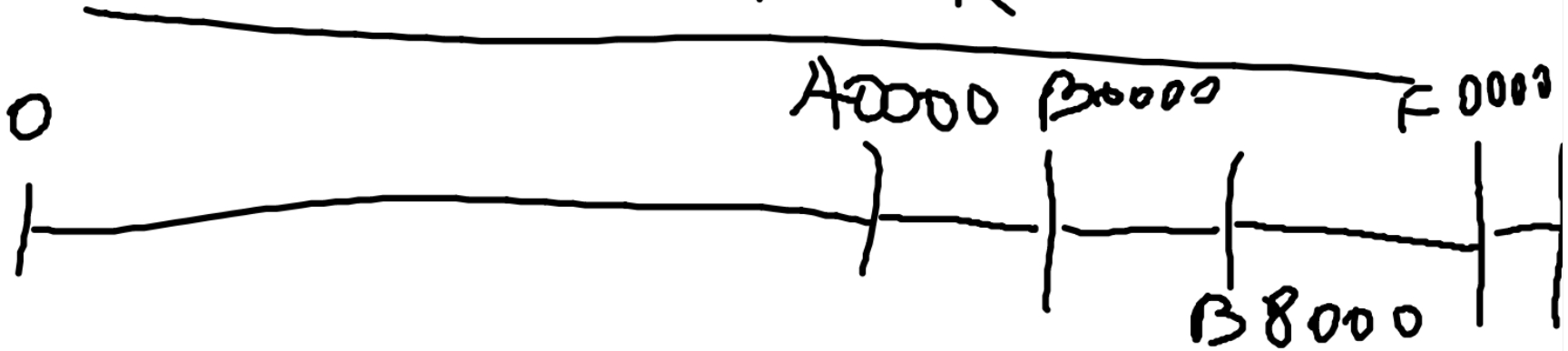
```
MOV AL, BL
AAM
MOV CL, AL
MOV DL, AH
OR DL, '0'
MOV AH, 2
INT 21H
MOV DL, CL
OR DL, '0'
INT 21H
```

```
CIKNOR: XOR   SI,SI
        MOV   AH,8
        INT   21H
        CMP   AL,0DH
        JZ    VEGNOR
        MOV   DL,AL
        MOV   AH,2
        INT   21H
        AND   AX,0000000000001111B
```

```
XCH   SI, AX
MOV   DX, 10
MUL   DX
ADD   SI, AX
JMP   CIKNOR
```

```
        MOV  DI, 10
        XOR  CX, CX
OSZCIK:
        XOR  DX, DX
        MOV  AX, SI
        DIV  DI
        PUSH DX
        INC  CX
        MOV  SI, AX
        OR   AX, AX
        JNZ  OSZCIK
```

```
        MOV AH,2
KLICIK:
        POP DX
        OR DL,'0'
        INT 21H
        LOOP KLICIK
```



FFFFØ

F000:FFFO AT

FFFF:Ø XT

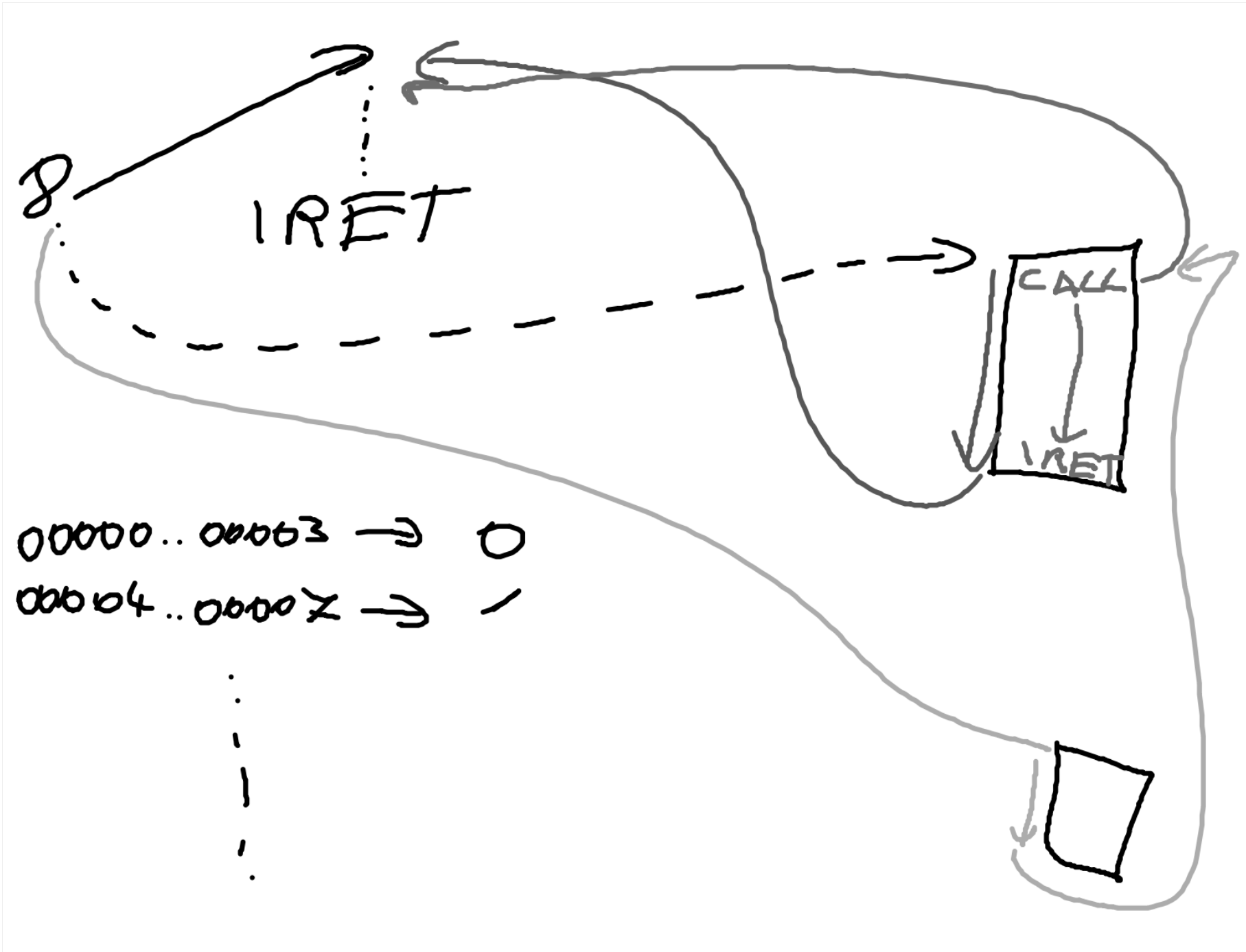

```
MOV  AX,0B800H
MOV  ES,AX
```

```
        MOV  CX, MENNY1
        MOV  SI, OFFSET S2
        XOR  DI, DI
CIKPAK:
        LODSB
        STOSB
        INC DI
        LOOP CIKPAK
```

```
S2      DB   'BALU FELSO'

MENNY1  EQU  $-S2

        MOV  AH,8
        INT  21H

        MOV  CX,MENNY1
        MOV  AL,' '
        XOR  DI,DI
TOCIKL:
        STOSB
        INC  DI
        LOOP TOCIKL
```

```
        MOV CX, 80*25
        XOR SI, SI
        XOR DI, DI
KECIK:  MOV AL, ES:[SI]
        ADD SI, 2
        CMP AL, 'V'
        JNZ NEMUCH
        INC BX
NEMUCH: LOOP KECIK
```

```
KECIK:  CMP BYTE PTR ES:[SI],'V'
        JNZ NEMVCH
        INC DI
NEMVCH: ADD SI,2
        LOOP KECIK
```


8
IRET
CALL
IRET
00000..00003 → 0
00004..0000Z → /

```
XOR   AX, AX
MOV   ES, AX

MOV   AX, ES:[8*4]
MOV   DX, ES:[8*4+2]
MOV   [REG1], AX
MOV   [REG1+2], DX

REG1  DW  2 DUP(?)
```

```
CLI
MOV     ES:[8*4], CS
MOV     ES[8*4+2], OFFSET SAYAT
STI
 :
 :
 :
```

```
SAYAT:  PUSH
        PUSH
          :
        POP
        POP
        JMP     DWORD PTR CS:[REG1]
```

INT 33H

AX ← "0"

"1" megjelenítés

"2" elrejtés

"3" állapotlekérdezés (X ← X, DX ← Y, BX ← gombok

INT 16H

"0" Leütés sváràs

"1" Leütés àllapot lekérdezés

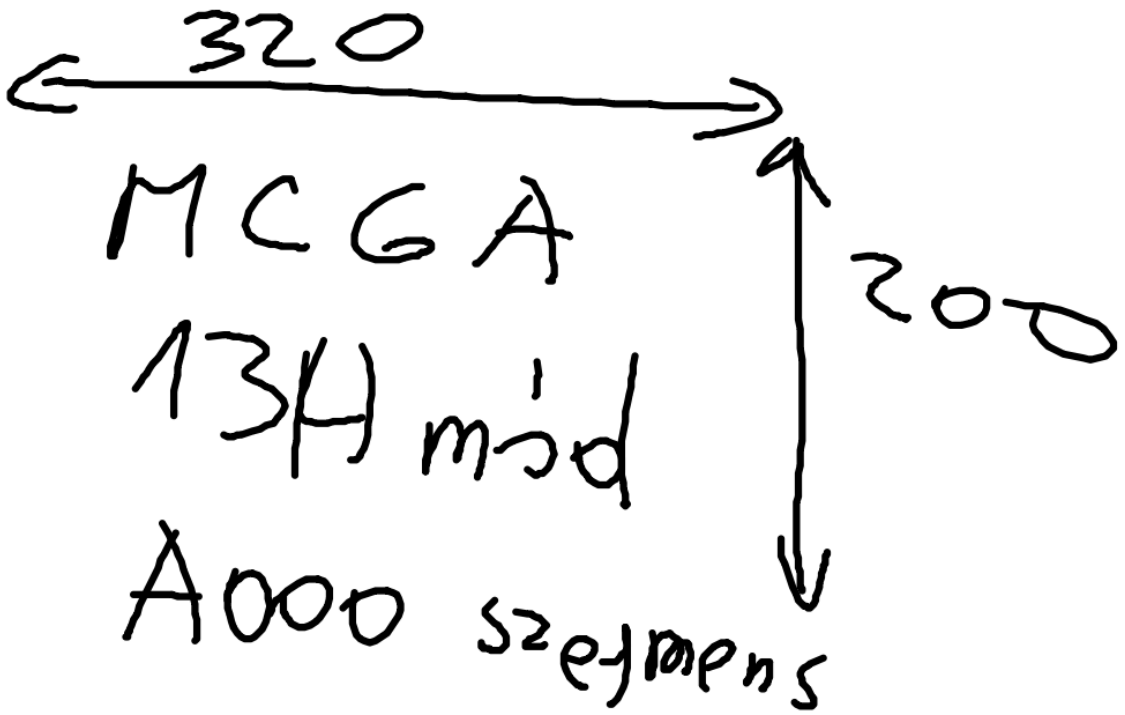
```
        MOV  AX, 1
        INT  33H
VAR:
        MOV  AX, 3
        INT  33H
        OR   CX, DX   / AND CX, NOT 7
        JZ   KILEP
        MOV  AH, 1
        INT  16H
        JZ   VAR
```

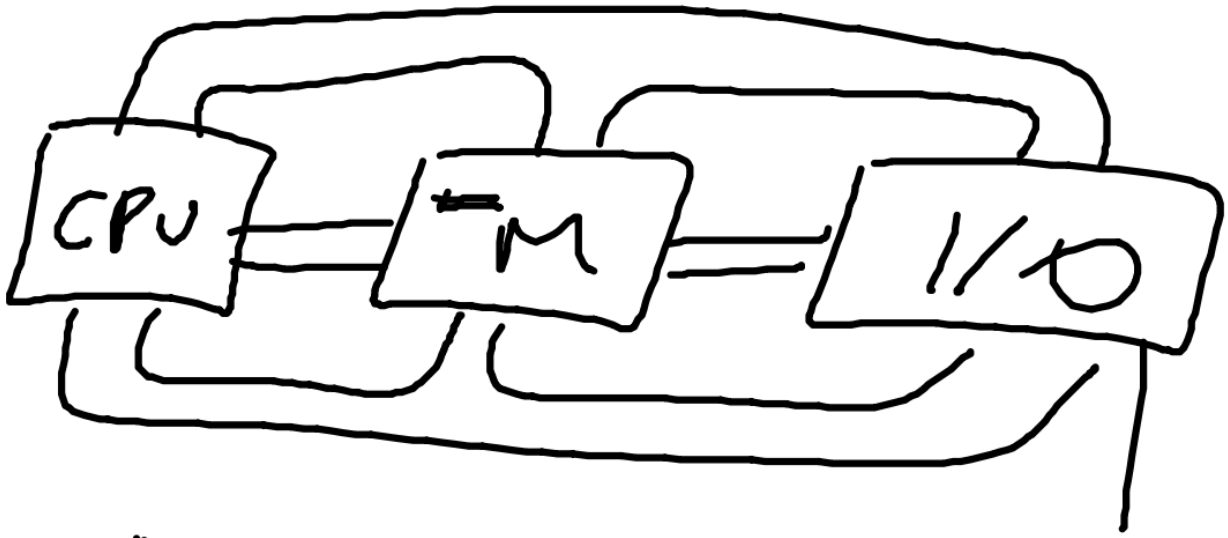
```
MOV  AH,0
INT  16H
CMP  AL,1BH
JNZ  VAR
```

KILEP:

```
MOV  AX,2
INT  33H
MOV  AH,4CH
INT  21H
```

```
MOV   CL,3
SHR   DX,CL
SHR   CX,CL
```





IN AL / AX) DIREKT DX ... [DX]

IN

OUT

OUT INDIREKT DX ... [DX]) AL / AH

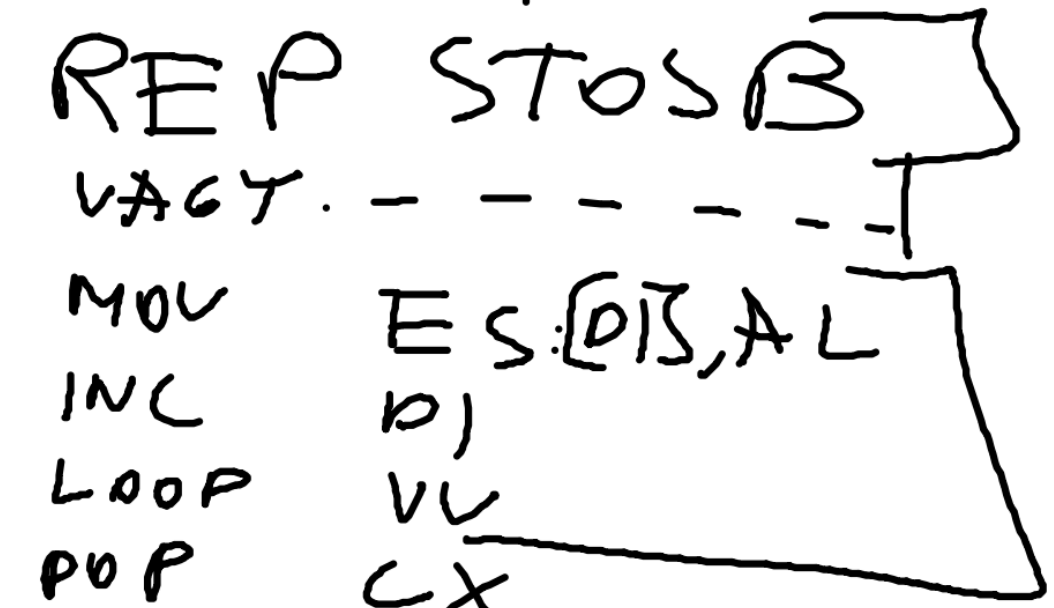
```
MOV DX 3C8H
MOV AL, 1
OUT DX, AL
INC DX
MOV AL, 63
OUT DX, AL
MOV AL, 0
OUT PX, AL
OUT DX, AL
```

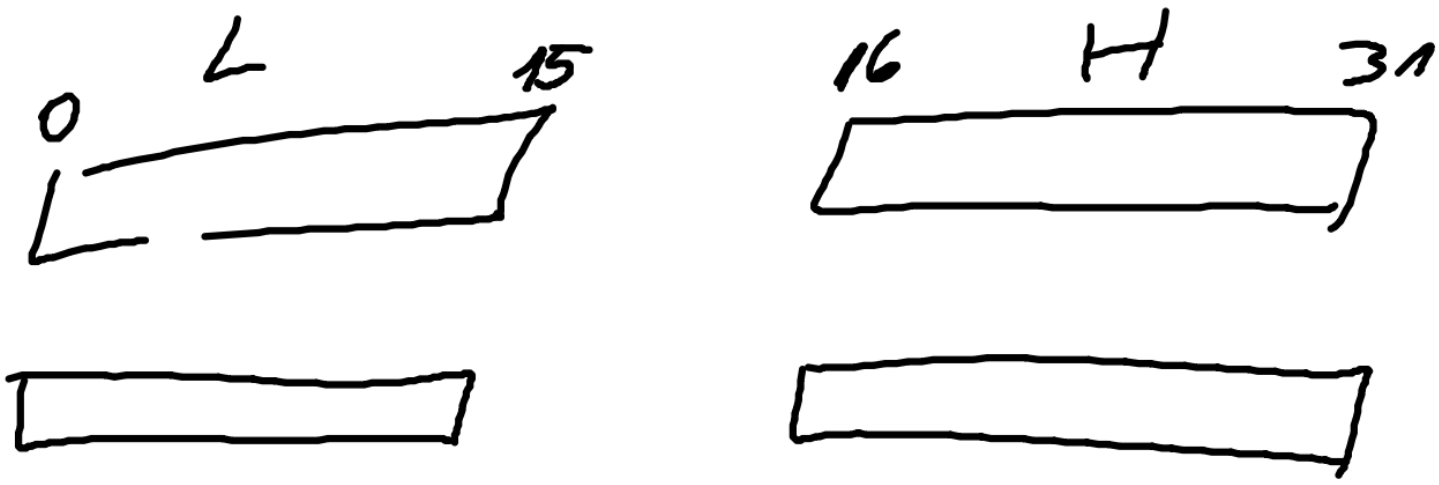
```
MOV  AX, 0A000H
MOV  ES, AX
MOV  AX, 13H
INT  10H
MOV  DX, 3C8H
MOV  AL, 1
OUT  DX, AL
INC  DX
MOV  AL, 63
OUT  DX, AL
MOV  AL, 0
OUT  DX, AL
```

```
OUT  DX,AL
DEC  DX
MOV  AL,2
OUT  DX, AL
INC  DX
MOV  AL,63
OUT  DX,AL
OUT  DX,AL
OUT  DX,AL
DEC  DX
MOV  AL,3
OUT  DX, AL
```

```
INC   DX
MOV   AL,0
OUT   DX,AL
MOV   AL,63
OUT   DX,AL
MOV   AL,0
OUT   DX,AL
MOV   DI,70*320+110
```

```
        MOV CX,20
        MOV AL,1
PICIK:  PUSH CX
        MOV CX,100
        REP STOSB
        VAGY.- - - - - -
W:      MOV ES:[DI],AL
        INC DI
        LOOP W
        POP CX
        ADD DI,220
        LOOP PICIK
```

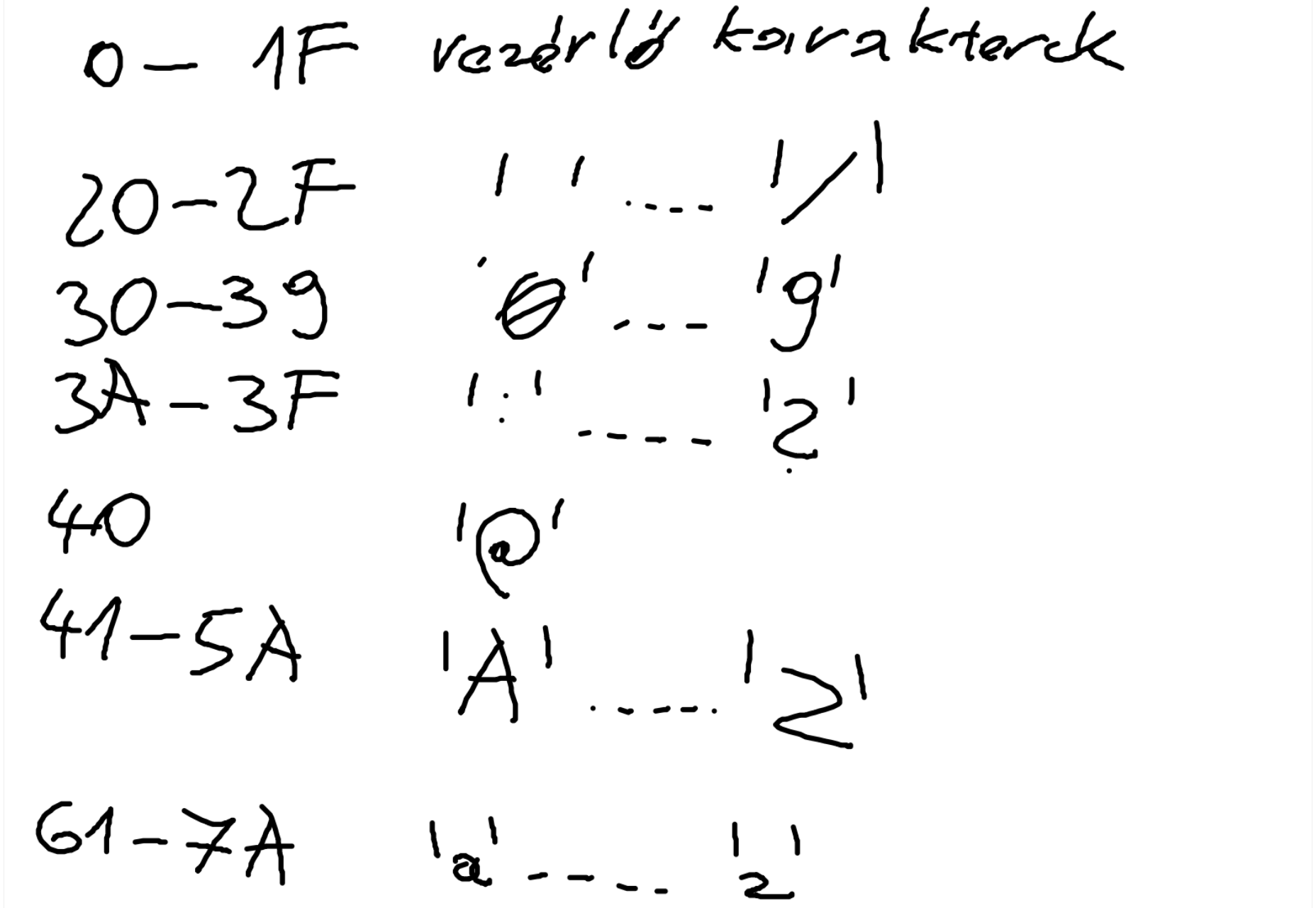




① ADD ② ADC

```
MOV  SI,80H
LODSB
OR   AL,AL
JZ   NUNCS
```

```
MOV DI,OFFSET SOKBYT
MOV CX,8000          MOV CX, 8000/2
MOV AL,25            MOV AX, (25*256)+25
REP STOSB            REP STOSW

SOKBYT  DB  8000 DUP(?)
```


0 – 1F vezérlő karakterek
20-2F ' ' ... '/'
30-39 '0' ... '9'
3A-3F ':' ... '?'
40 '@'
41-5A 'A' ... 'Z'
61-7A 'a' ... 'z'

X X X X X X X X

0 1 0 0 0 0 0 1

41H