

# II konzultáció

2006. Dec. 2.

Soros kommunikáció

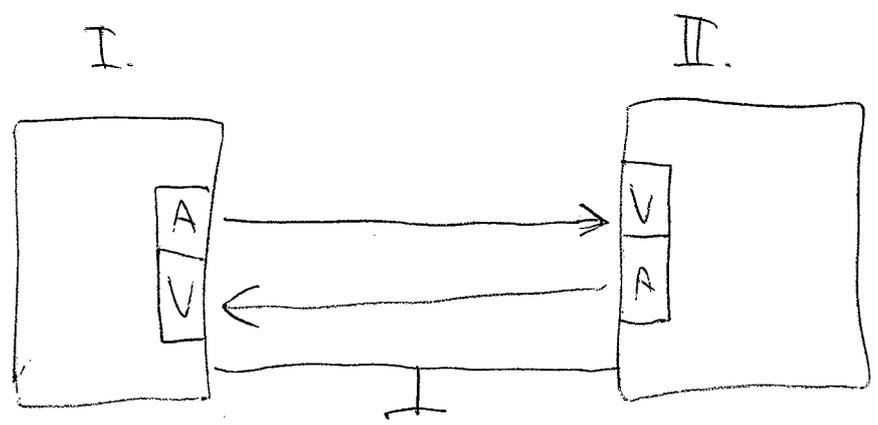
## LABOR UT MUTATÓ!!!

PROGRAMJA IT NÉZD  
A'T, MERT A2 LESZ  
NYÁRON, MOST MEG HASONLÓ!

fellegre TOTÁL illyenek  
kenned!!! Vagy akár egy  
is lehetnek ~~egy~~ most csigán!!!

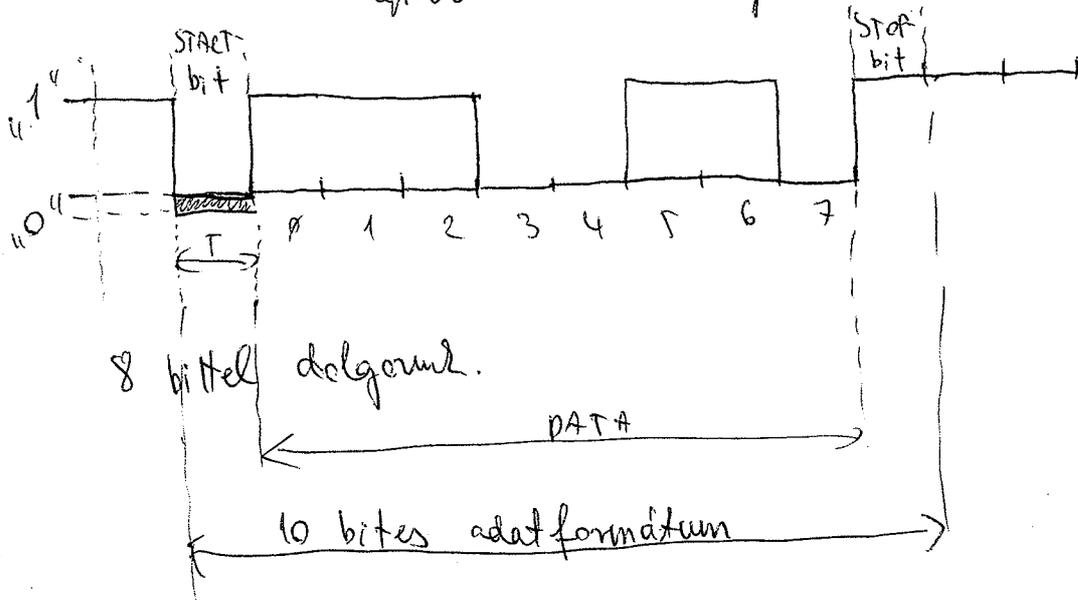
Soros kommunikáció!

- egyirányú → SIMPLEX
- fél-duplex
- duplex



Ha nincs csigán: aszimmetrikus kommunikáció  
 Ha van: szimmetrikus -||-

Amirshani az adat és a vtel fgggetlen!  
 a legtbls kntroller ilen.



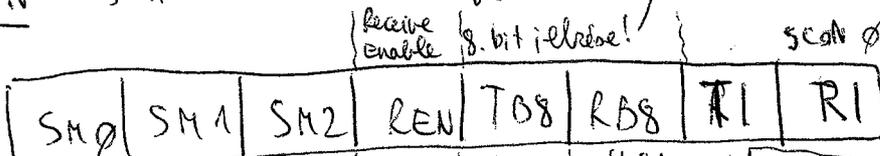
TXD → adas  
 RXD → vtel

Tx Clock → PortB  
 Rx Clock → PortB  
 P3.0;  
 P3.4!

11 bit esler lehet plumbus, paritassbit, jelssbit, stb.

plumbus:  $0110 | 0111 | B = 67_H$

SCON: Serial Control Register



SBUF: Vteli: csak OLVASHATÓ!

Serial Buffer

Adási: csak ÍRTHATÓ!

MODE	SM0	SM1	formátus
0	0	0	8 bites vtel üzemmod
1	0	1	→ 10 bites vtel üzemmod; 11 bites form/32v64
2	1	0	11 bites vtel üzemmod; 11 bites form/32v64
3	1	1	11 bites vtel üzemmod; 11 bites form/32v64

"0": SM2: 10bits üzemmod

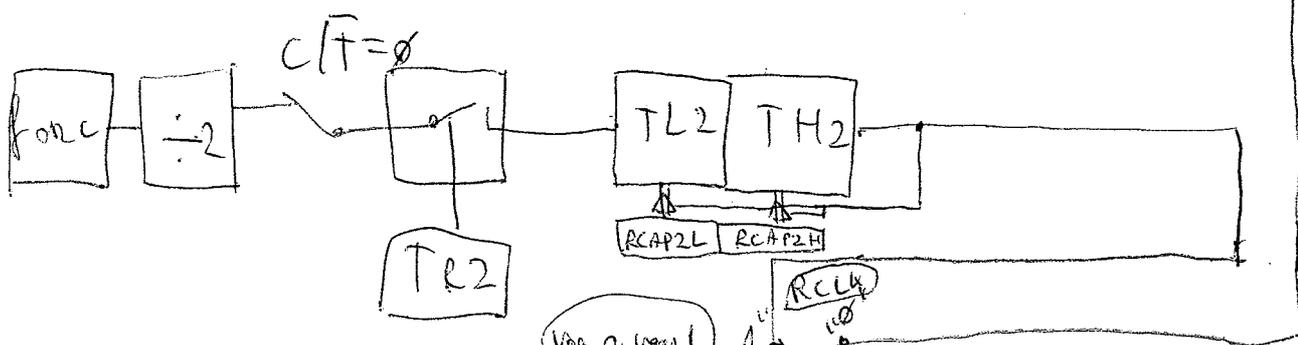
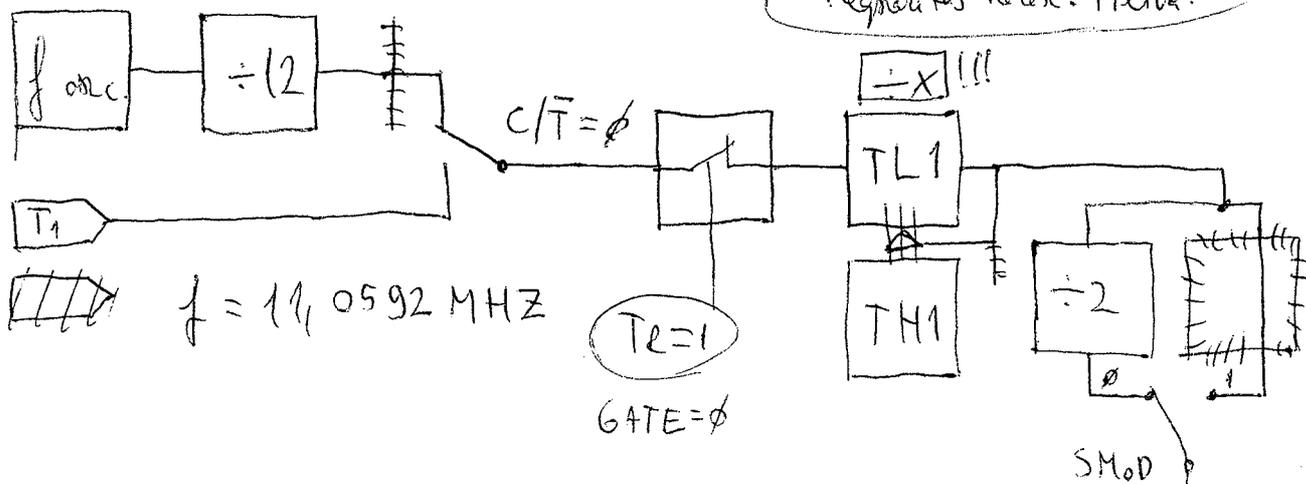
az SM2 lpteti be a

STOP-bitet (1-es nrtjel)

"1": multiprocessoros üzemmod!

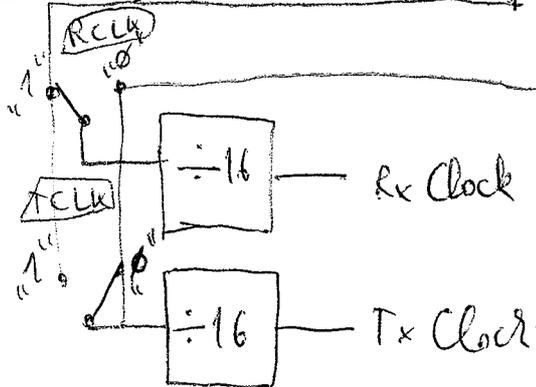
paramos vagy byte iön?

# 1-es időzítő / mérő



baud-rate:  $\frac{f_{osc}}{12} \cdot \frac{1}{X} \cdot \frac{2^{SMod}}{2} \cdot \frac{1}{16}$

↑  
8 bites mérték-érték



ebből az x:

$$X = \frac{f_{osc} \cdot 2^{SMod}}{12 \cdot 2 \cdot 16 \cdot \text{Baudrate}}$$

Helyes értéket adja-e a megoldás?

$2^8 - X$  : újratöltési érték

1200 baud érték  $X = ?$   $S_{mod} = 1$  érték

$$X = \frac{11,052 \cdot 10^6}{12 \cdot 16 \cdot 1200} = \underline{\underline{48}} \text{ - az órára van mérve!}$$

$$2^8 - X = 256 - 48 = \underline{\underline{208_0}} \text{ az újatöltési érték!!!}$$

Teljes:  $TL_1 = 208_0$   
 $TH_1 = 208_0$

Ha  $S_{mod} = 0 \Rightarrow$   
 $TL_1 = 232$   
 $TH_1 = 232$

232 az újatöltési érték!!!

Nyilván, mert  $S_{mod} = 0$  bekapcsolja a 2-es órákat.

Direct címzés:

MOV dir, #data

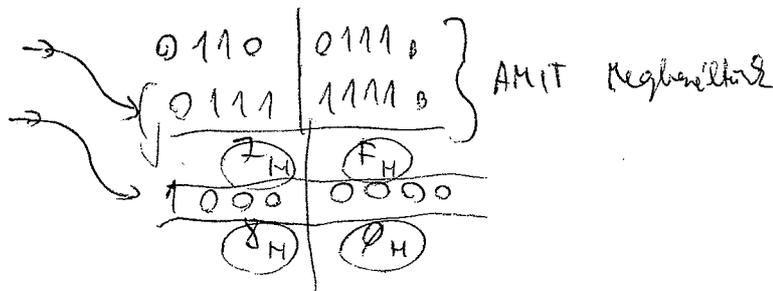
MOV TH1, #232D

ANL PCON, #7FH

ORL PCON, #80H

Nem bit címezheti a PCON regisztert!

$\rightarrow$  használható decimálisan is!



INIC -nél 3 dolgot kell beállítani (belső interfésznél):

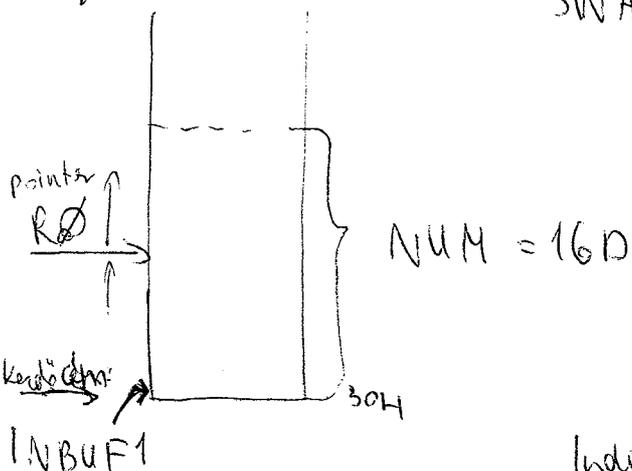
- Megnyitni az órákat
- Időzítőt / mérőidőt
- Soros interfészt

# PELDA'K

Memória-feltöltés : Labor - segédlet [I/2]-s fellda

I/2

SWA = FD 00 H



```

cel formás
MOV @R0, A
    
```

Indirect cím: @R<sub>i</sub>      i = 0, 1

R<sub>0</sub> len a pointer => növegy!

Célszerűbbé

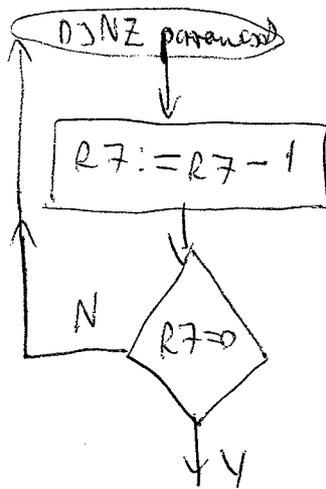
```

DJNZ R7, rel
DJNZ dir, rel
    
```

Decrement and Jump if Not Zero

```

DJNZ R7, VISSZA
    
```



(FILL 1)

Adat feltöltés

Mentés -> Stack  
ACC, PSW, DPH, PPL

R7 := NUM  
R0 := INBUF1

DPTR := SWA

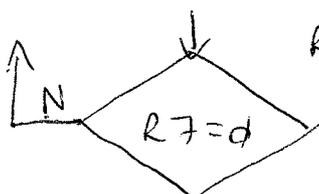
A := {DPTR}

{R0} := A

R0 := R0 + 1      ← pointer-növelés

R7 := R7 - 1

R0 - 0 EQU 00H -> mentés



Visszaállítás a Stack-ből  
DPL, DPH, PSW, ACC

RET

Ennél a formájelnél ASSEMBLY programja!

FILL1: ; A művelet megkezdése  
 INBUF1 EQU 30H  
 NUM EQU 16D  
 SWA EQU FD00H ; A φ kell elő, hogy tudjuk ⇒ Ez nem!!!

PUSH ACC  
 PUSH PSW  
 PUSH DPH  
 PUSH DPL

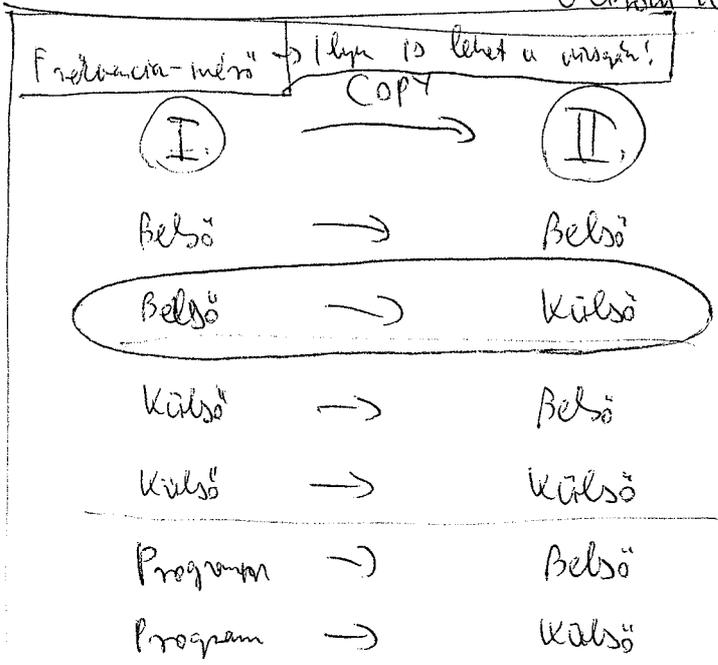
MOV R7, #NUM ; a # kódszám!!! R7=16D  
 MOV Rφ, #INBUF1 ; Rφ=30H  
 MOV DPTR, #SWA ; DPTR=FD00H  
 MOVX A, @DPTR ; X → külső memóriába mov!

VISSZA1: MOV @Rφ, A ; ←  
 INC Rφ  
 DJNZ R7, VISSZA1  
 POP DPL  
 POP DPH  
 POP PSW  
 POP ACC  
 RET

Kém

M0 V007, # data

→ először kihagyhatom az ACC-t, és csak  
behozom az adatot és kérem.



EZ EGY COPY program!

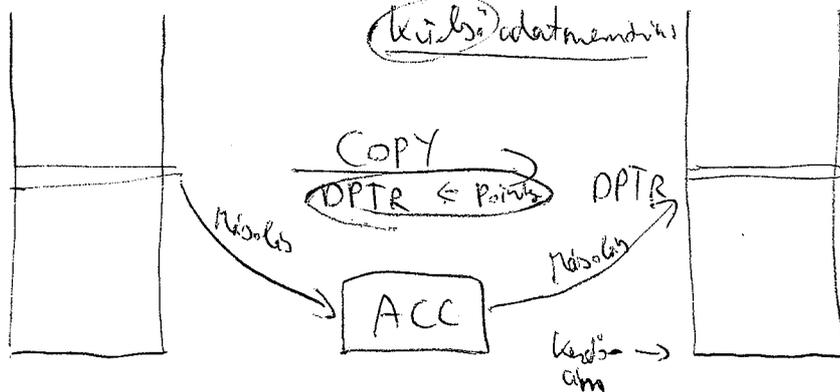
Memória-másolás lehetőségei

Belső adatmemória

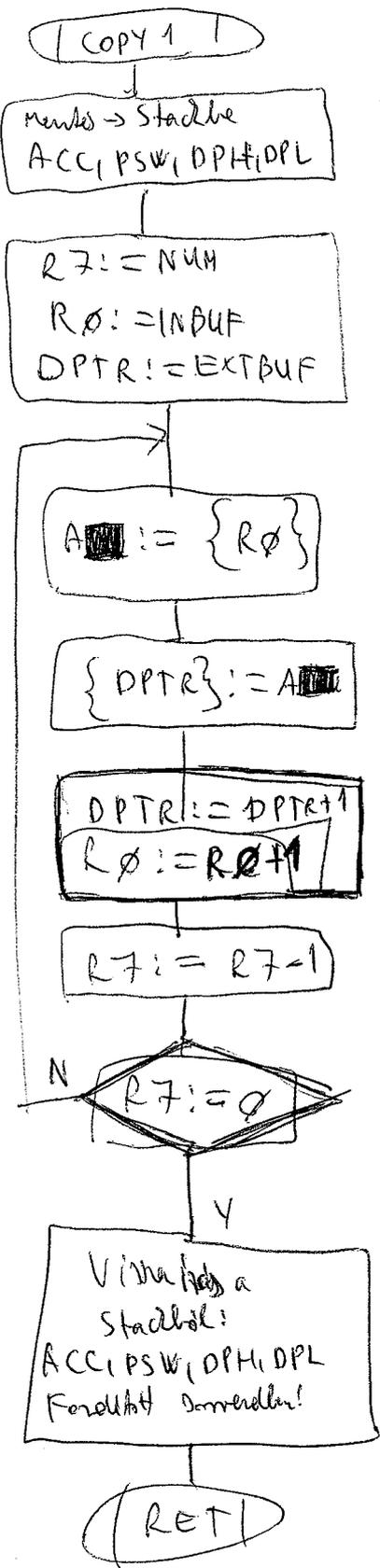
Külső adatmemória

Code or R0, R1 lehet ← Pointer → R0

Kérdő cím



R7: Cirkus mávaló



COPY 1:

```

    INBUF EQU 30H
    NUM EQU 160
    EXTBUF EQU 9000H

    PUSH ACC
    PUSH PSW
    PUSH DPH
    PUSH DPL
    MOV R7, #NUM
    MOV R0, #INBUF
    MOV DPTR, #EXTBUF
  
```

VISSZA 1:

```

    MOV A, @R0
    MOVX @DPTR, A
    INC DPTR
    INC R0
    POP DPL
    POP DPH
    POP PSW
    POP ACC
    RET
  
```

**Figyelem!** ACC → műveleti utasítás  
 A → műveleti kód

Egyébelső egyrészt állásúna PELDA: Szoftver tesztelési művelet.

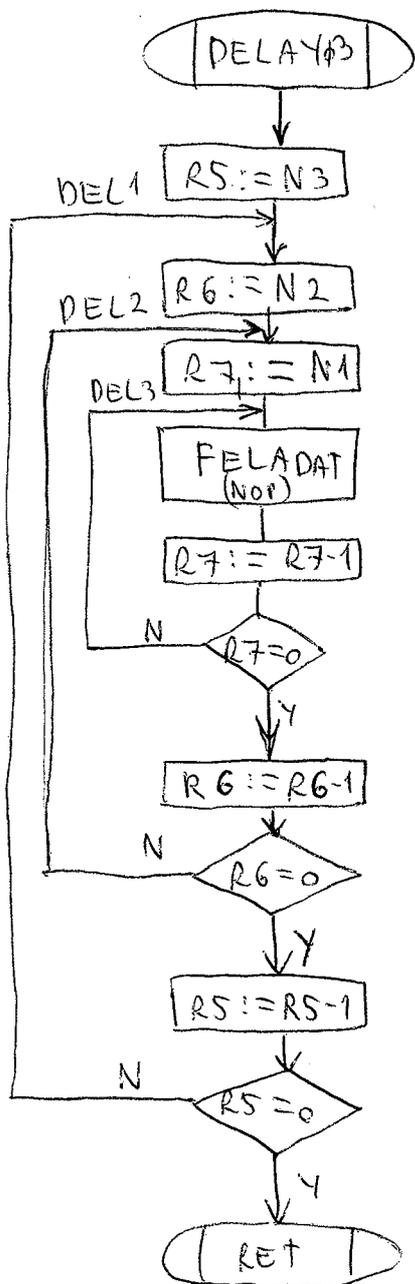
Feladat:  $N1 * N2 * N3$      $N1: R7-es$      $N2: R6-os$      $N3: R5-os$  regiszterbe  
 from!

NOP: Not OPERATION  
 MOV R7, #data  
 DJNZ R2, rel  
 RET

} NOP: egy gépi utasítás  
 } ÖMÉ - címre ilyen utasítás van a program!  
 } Ahol (MOV, DJNZ, RET)  $\geq$  gépi utasítás.

FA:

ASSEMBLY:



DELAY 3:

MOV R5, #N3

DEL 1: MOV R6, #N2

DEL 2: MOV R7, #N1

DEL 3: NOP

DJNZ R7, DEL 3

DJNZ R6, DEL 2

DJNZ R5, DEL 1

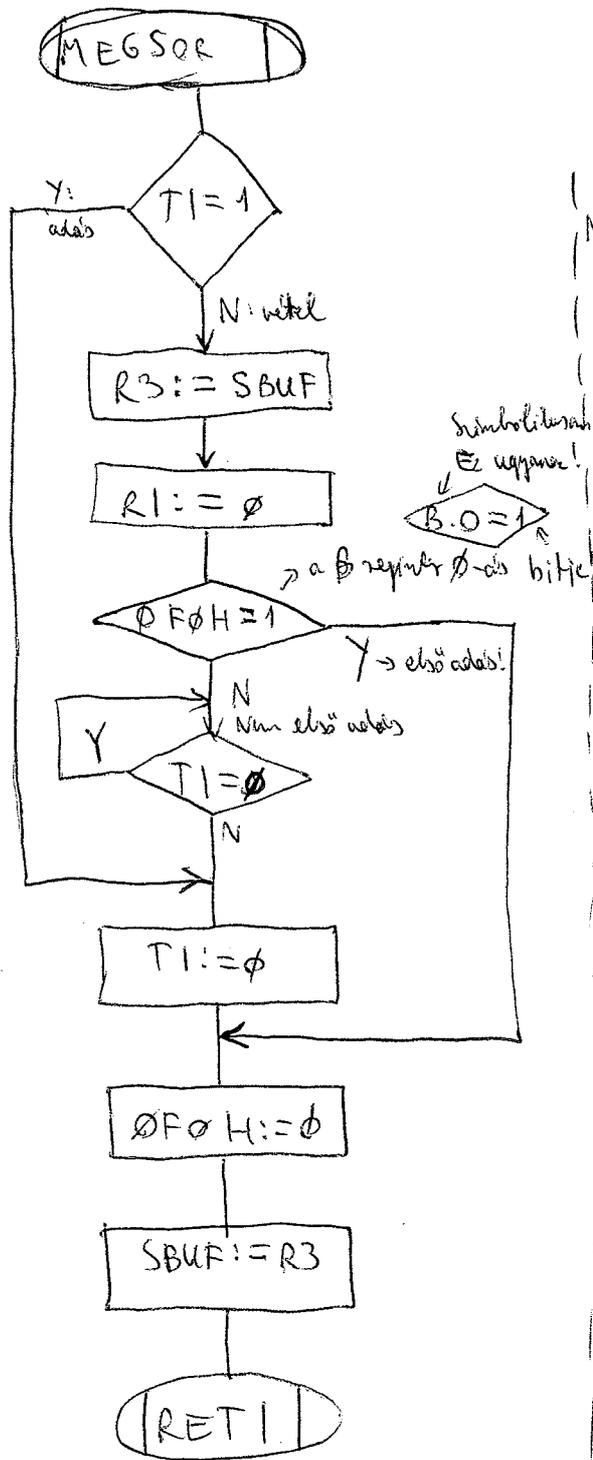
RET

MEGSOR: Megszakítás - rutina (label átmenet!)

(label Interrupt System mellélet!)

ORG 23H

LJMP MEGSOR



MEGSOR:

JB TI, ADAS

MOV R3, SBUF

CLR R1

JB 0F0H, E\_ADAS

JNB TI, VAR

VAR:

E\_ADAS: CLR 0F0H

ADAS: MOV SBUF, R3

CLR TI

RETI

Most BITET kell törölni!!!  
Hopp! rá!

Mi is a különbség a RET és a RETI között? Nézz utána!

SER: Ebből a mellékelt lepon! (Labor-utmutató!)

DISPADR: Megint miből, de nem használható vizsgán!

Helyes: MOV DPTR, #DISP BUF

Mi az a T bit? Nezz utána!

MOV SBUF, #CR

CR EQU 0DH

VAR\_CR JUMP TI, VAR\_CR

CLR TI

VIZSGA IDŐSZAK: Eddig rendelkezés 13<sup>45</sup> - 15<sup>45</sup> -ig van telefonügyelet!  
Időpontot írt vanál a NEPTUNON!

SZÁMOLDGÉPET hozz!!! INICHEZ kell!

COPY lea (mire kutas)!