Fealadat4: param.asm

Feladat meghatározása

A program célja bemutatni a parancssoros paraméterátadást. A hallgatók megismerik a PSP felépítését, szerepét, valamint annak elérését. A program az előző programhoz hasonlóan egy golyó leesését szimulálja. Paraméterként megadható a golyó vízszintes irányú kezdő-sebessége. Ekkor a test egy parabola pályán mozog, amit a program kirajzol. Paraméter nélkül indítva a programot, az egy alapértelmezett (10m/s) kezdősebességű testet szimulál. A számítás alapja:

$s_y = ½ * g * t^2$

$s_x = v_x * t$

Program bemutatása:

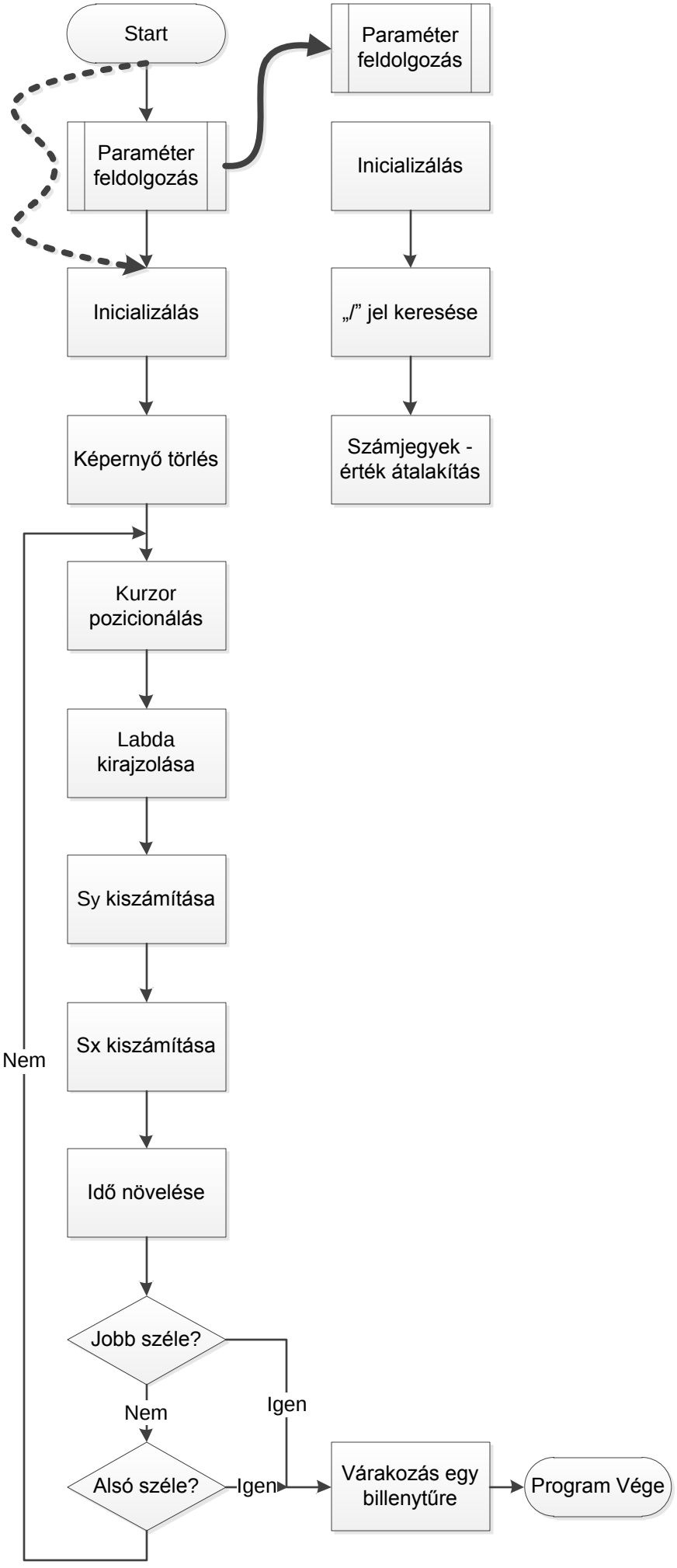
- A programot először a bemenő paramétereket feldolgozó rész nélkül kell megírni. Ekkor a program az alapértelmezett 10m/s kezdő-sebességgel induló test pályáját rajzolja ki.
- A működő programot ki kell egészíteni a paraméter feldolgozó résszel. Az így megírt program szigorúan „/11” formátumban fogadja a paramétereket. Önálló gyakorlásként ajánlható a paraméter feldolgozás hibakezelésének meg-írása. Például a program ne számoljon a „/w2” paraméterrel.

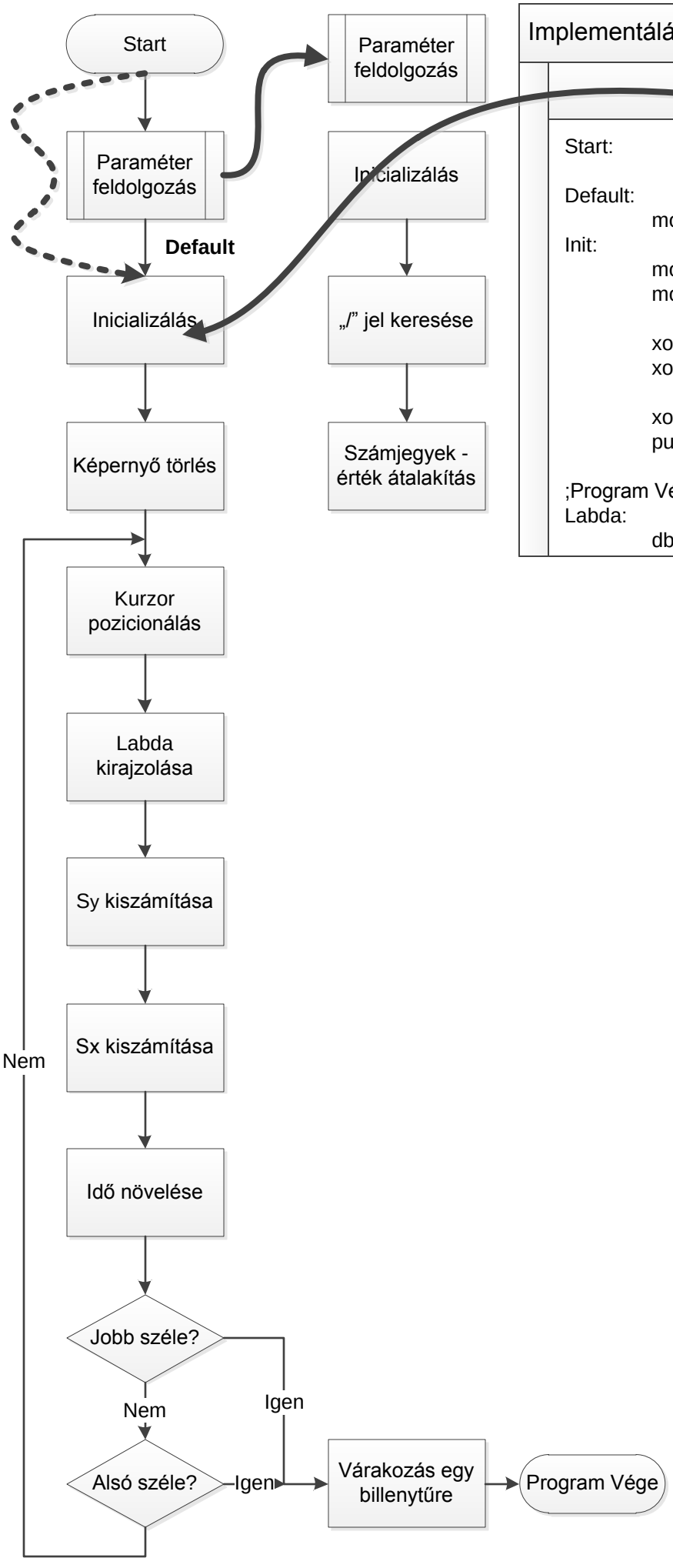
Implementálás

```
Code    Segment
        assume CS:Code,
        DS:Data, SS:Stack
Start:
        mov     di, 82h

        mov     cx, 10
Keres:
        mov     dl, [di]
        cmp     dl, "/"
        jz      ParamKezdet
        inc     di
        loop    Keres
        jmp     Default

ParamKezdet:
        inc     di
        mov     bl, [di]
        sub     bl, 48
        inc     di

        mov     bh, [di]
        sub     bh, 48

        mov     ax, 10
        mul     bl

        add     al, bh
        mov     cx, ax

        jmp     Init

Default:
        mov     cx, 10

Init:
        mov     ax, Code
        mov     ds, ax

        xor     di, di
        xor     si, si

        xor     dx, dx
        push    dx

Torles:
        mov     ax, 03h
        int     10h

Rajz:
        mov     bx, di
        mov     dh, bl
        mov     bx, si
        mov     dl, bl
        xor     bh, bh
        mov     ah, 02h
        int     10h

        mov     dx, offset Labda
        mov     ah, 09h
        int     21h
```

Implementálás

```
        pop     ax
        push    ax

        mov     bl, al
        mul     bl
        shr     ax, 1
        mov     di, ax

        pop     ax
        inc     ax
        push    ax
        dec     ax

        mov     bl, cl
        mul     bl
        mov     si, ax

        cmp     si, 80
        jnc     Var

        cmp     di, 25
        jnc     Var

        jmp     Rajz
Var:
        xor     ax, ax
        int     16h

Program_Vege:
        pop     cx
        mov     ax, 4c00h
        int     21h

Labda:
        db      "O$"

Code    Ends

Data    Segment
Data    Ends

Stack   Segment
Stack   Ends

        End Start
```

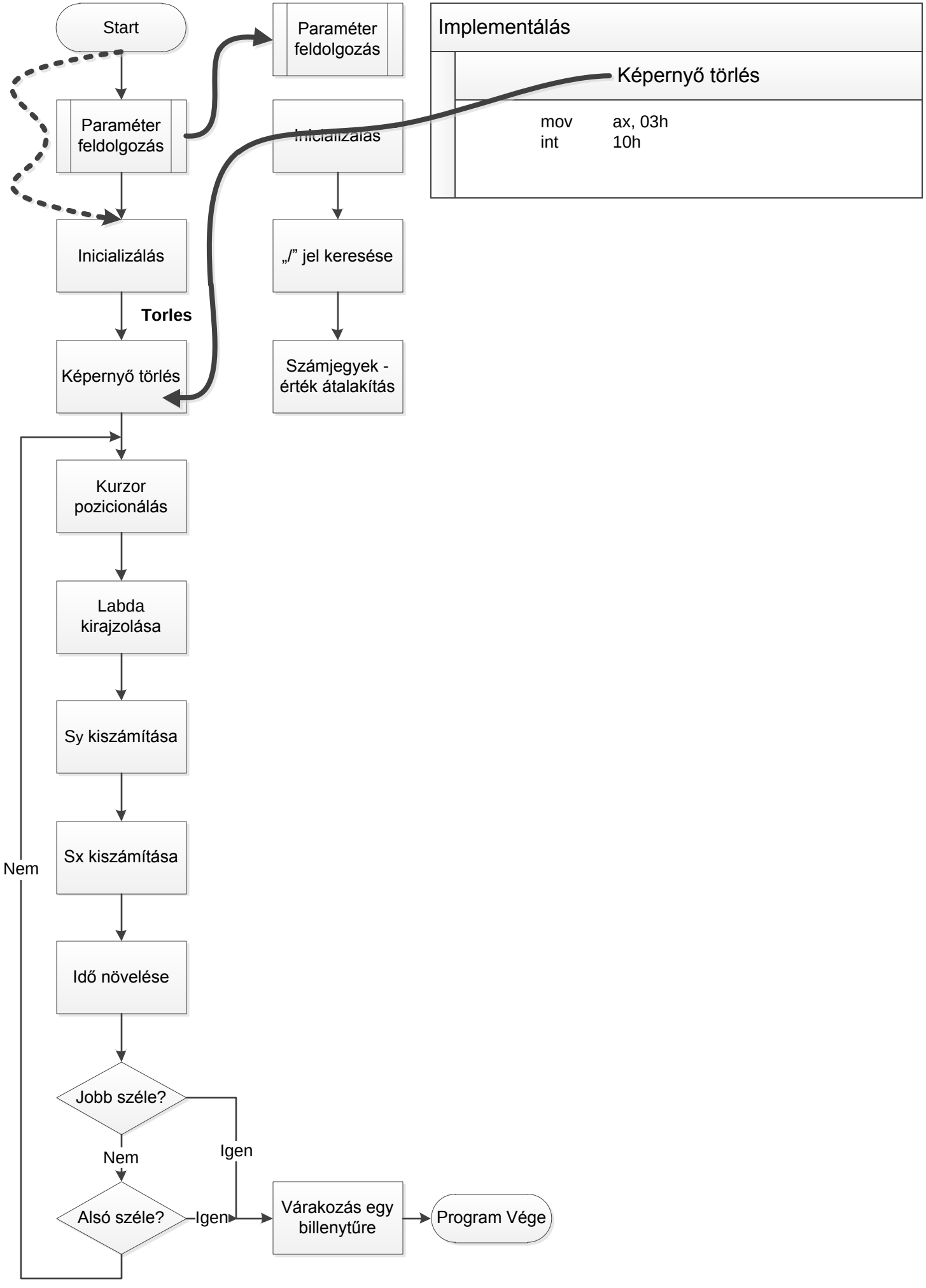

Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Képernyő törlés
Számjegyek - érték átalakítás
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Nem
Idő növelése
Jobb széle?
Igen
Nem
Alsó széle?
Igen
Várakozás egy billenytűre
Program Vége

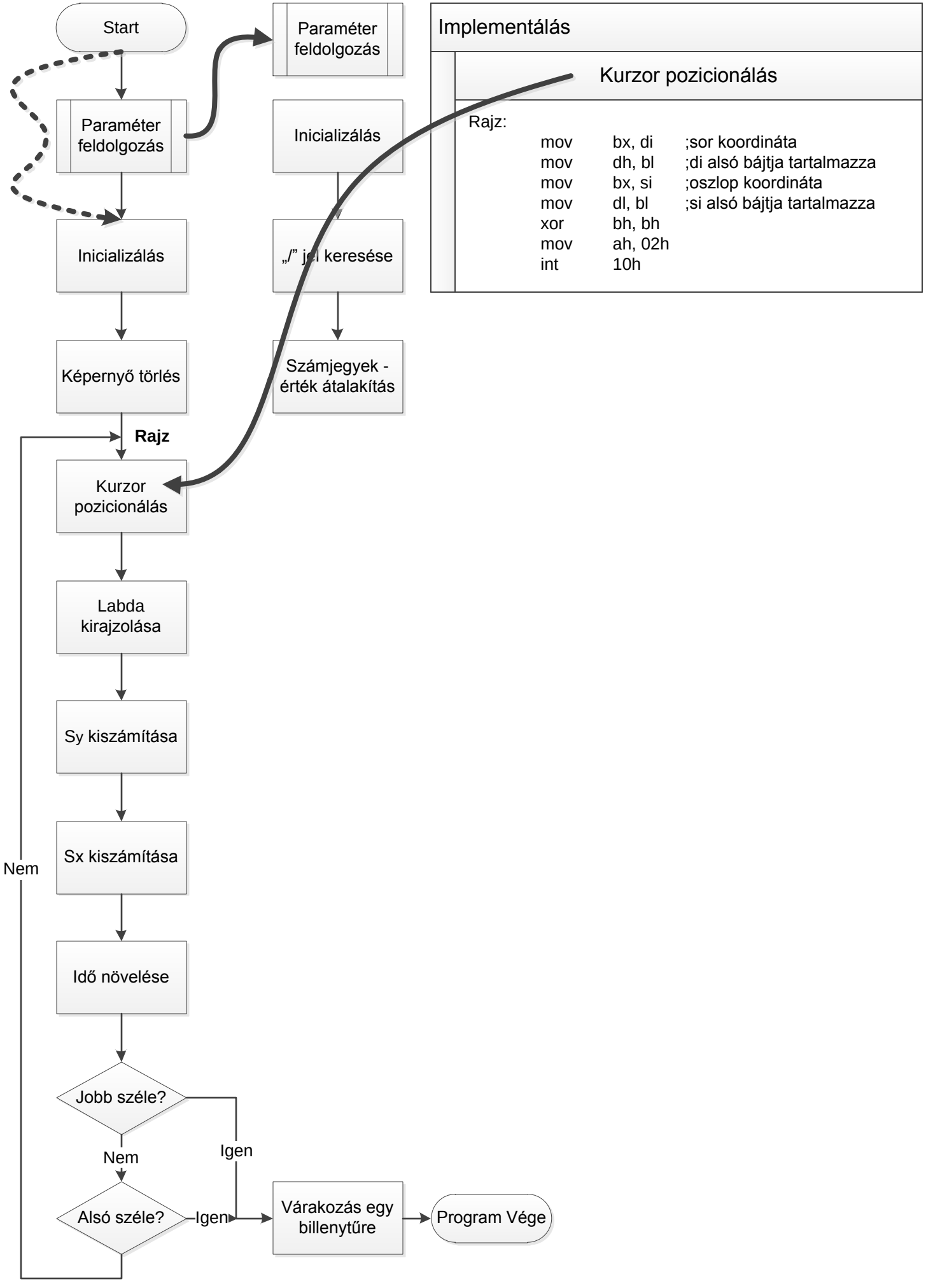


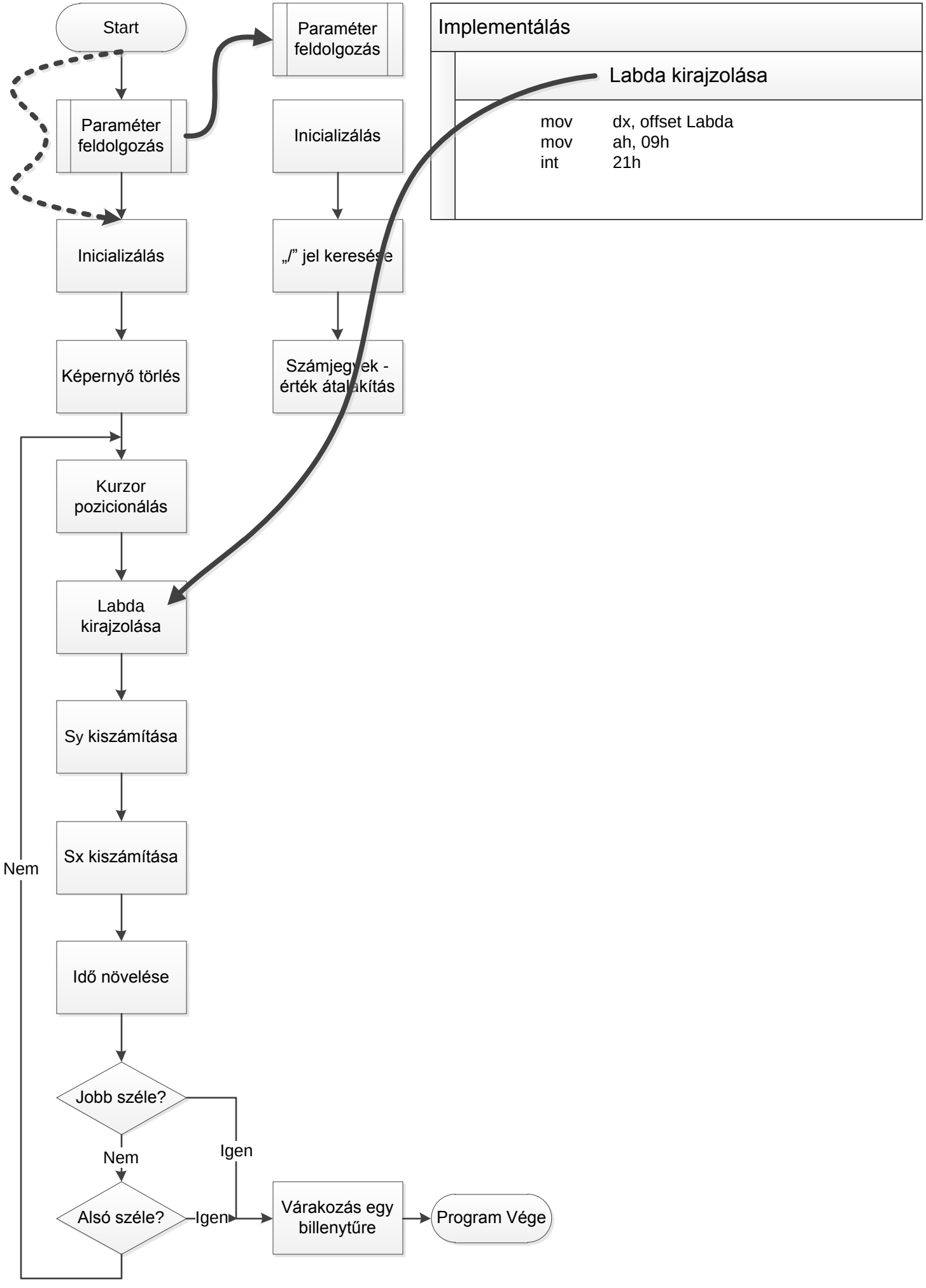
Implementálás

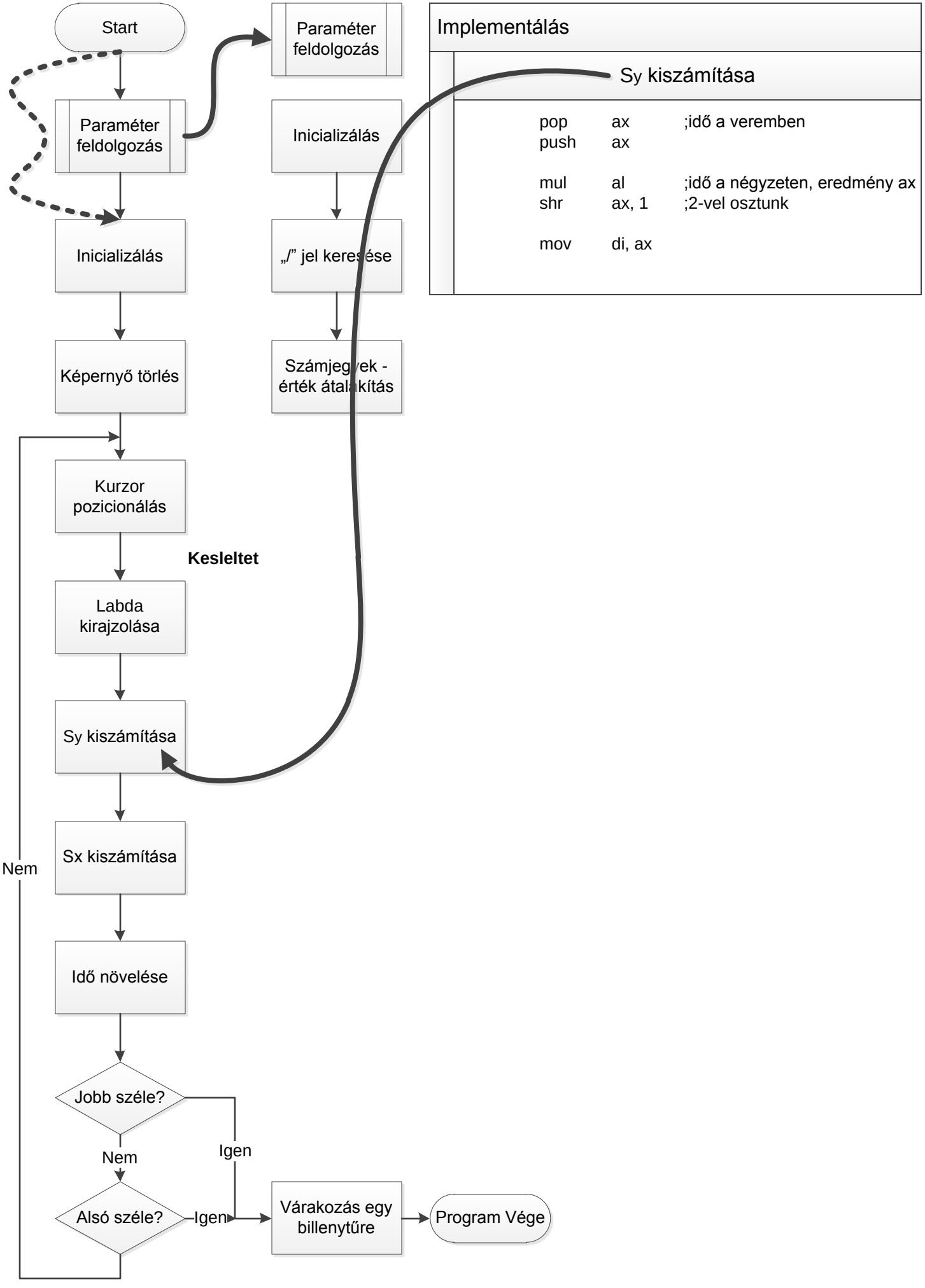
Inicializálás

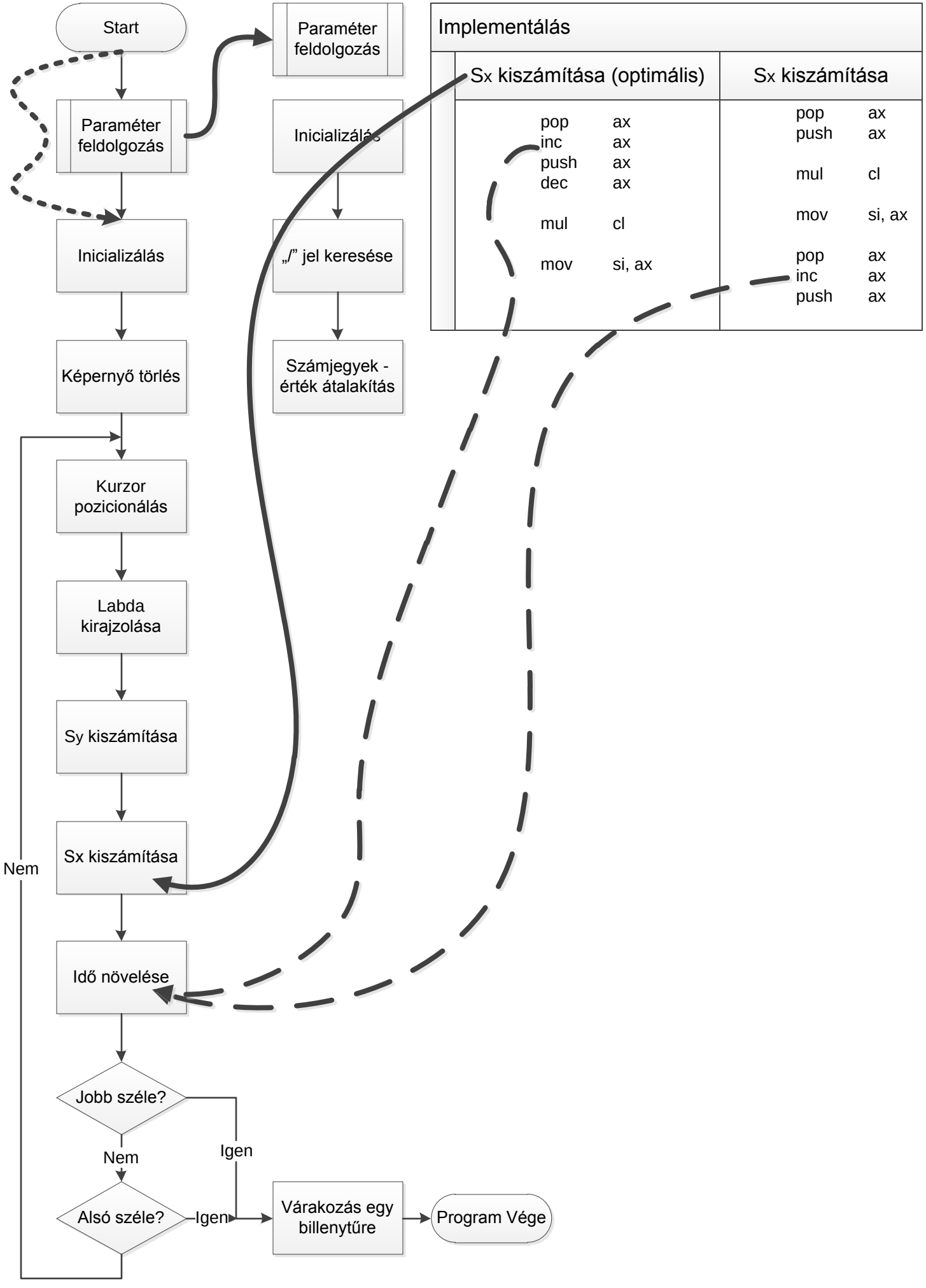
```
Start:

Default:
        mov     cx, 10      ;Alapértelmezett v0
Init:
        mov     ax, Code
        mov     ds, ax

        xor     di, di      ;golyó sor pozíciója
        xor     si, si      ;golyó oszlop pozíciója

        xor     dx, dx
        push    dx          ;verembe az idő (most 0)

;Program Vége után, de még a Code szegmensbe
Labda:
        db      ”0$"
```

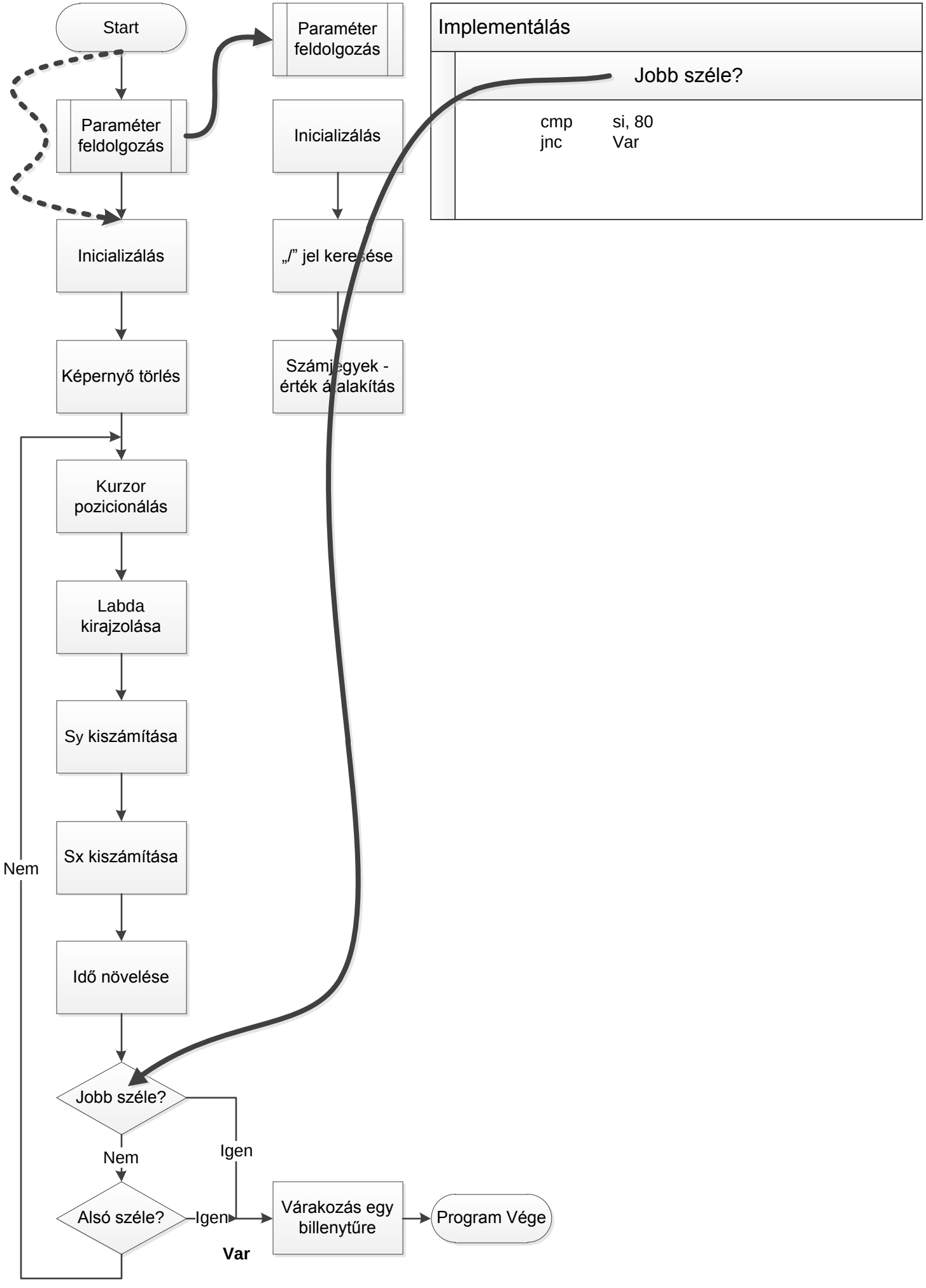

Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Számjegyek - érték átalakítás
Torles
Képernyő törlés
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Idő növelése
Jobb széle?
Nem
Igen
Alsó széle?
Igen
Nem
Várakozás egy billenytűre
Program Vége
Implementálás
Képernyő törlés
mov ax, 03h
int 10h

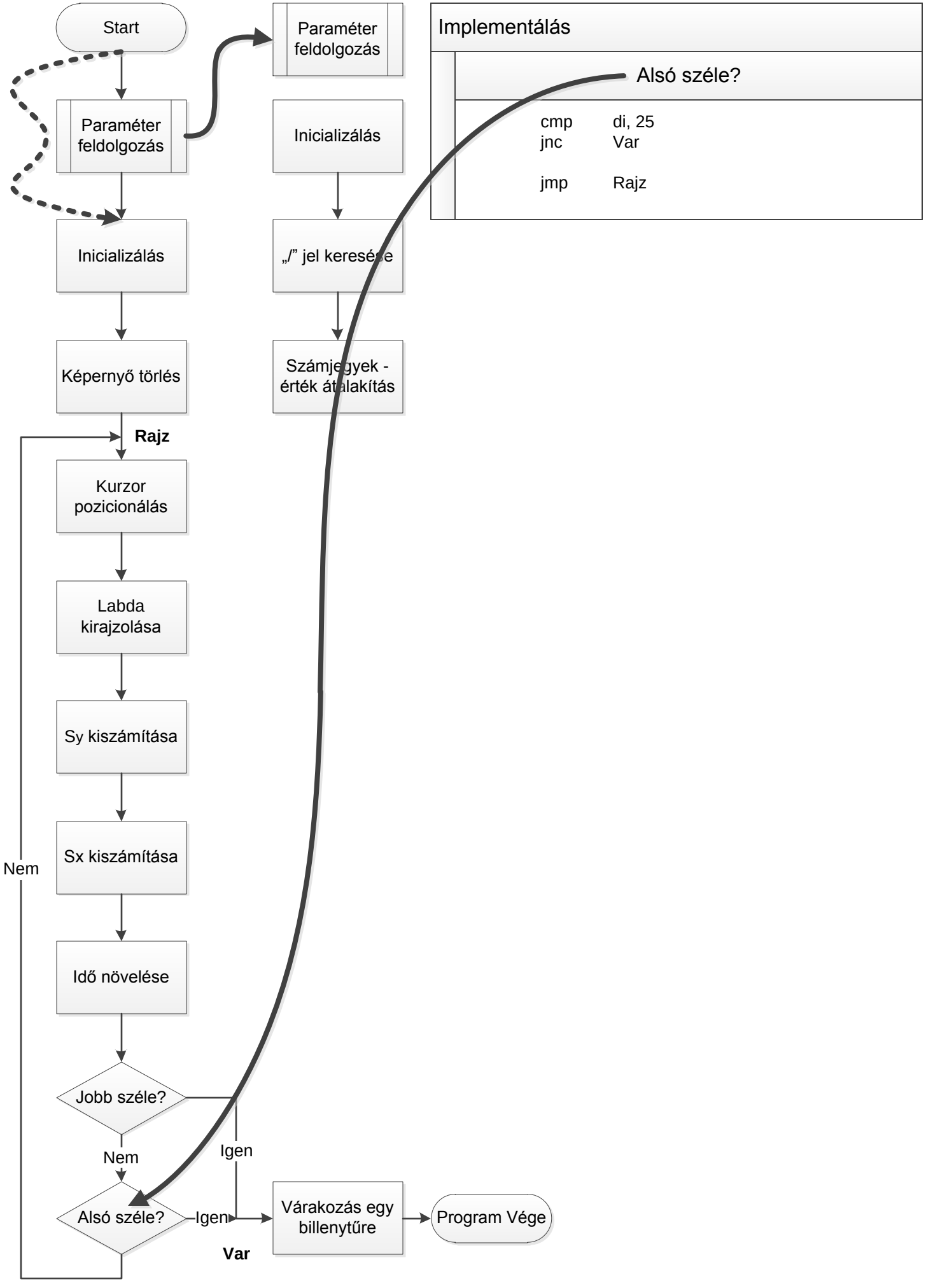

Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Képernyő törlés
Számjegyek - érték átalakítás
Rajz
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Idő növelése
Jobb széle?
Nem
Igen
Alsó széle?
Igen
Várakozás egy billenytűre
Program Vége
Nem
Implementálás
Kurzor pozicionálás
Rajz:
mov bx, di ;sor koordináta
mov dh, bl ;di alsó bájtja tartalmazza
mov bx, si ;oszlop koordináta
mov dl, bl ;si alsó bájtja tartalmazza
xor bh, bh
mov ah, 02h
int 10h

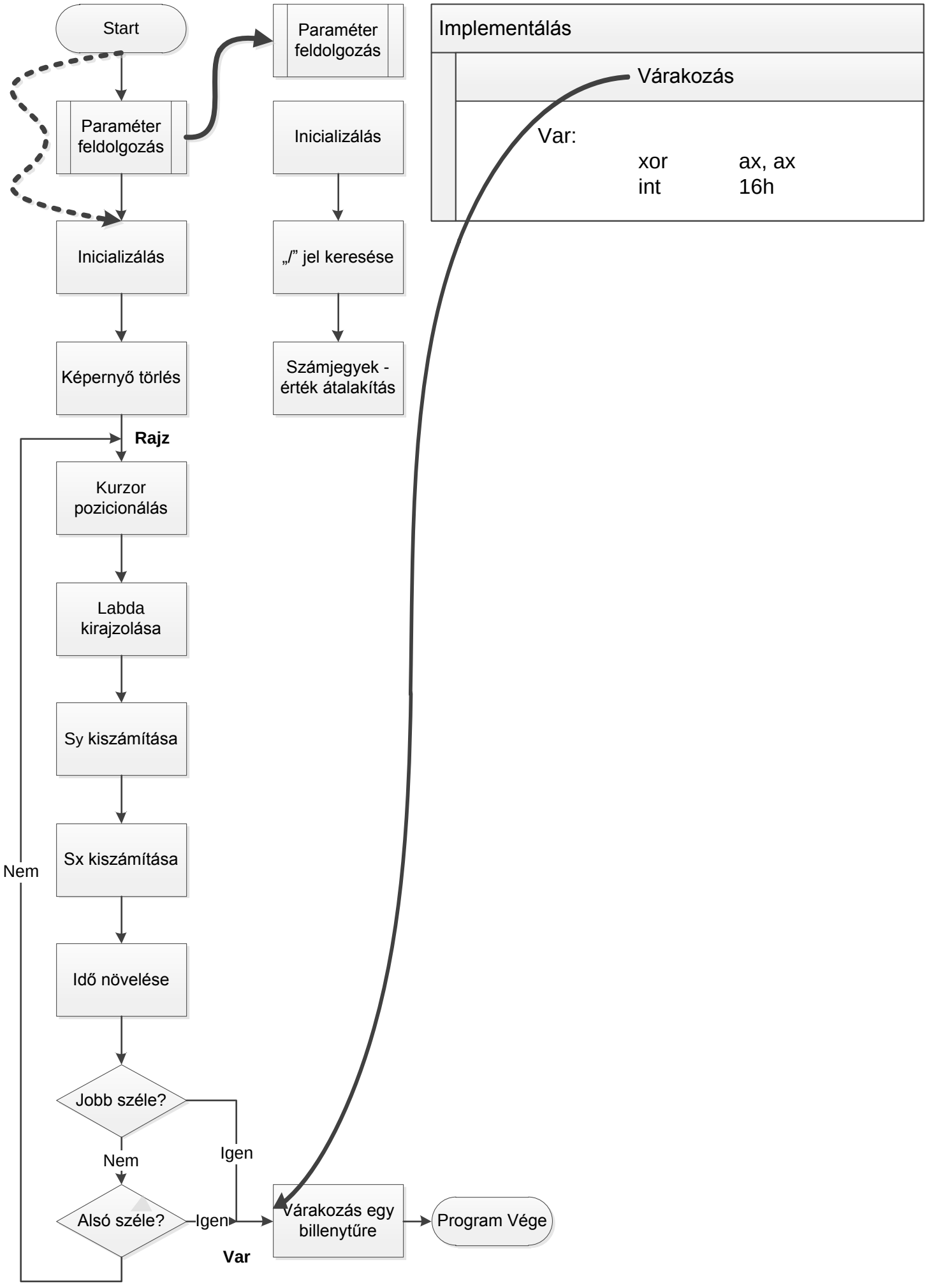

Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Képernyő törlés
Számjegyek - érték átalakítás
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Idő növelése
Jobb széle?
Nem
Igen
Alsó széle?
Igen
Várakozás egy billenytűre
Program Vége
Nem
Implementálás
Labda kirajzolása
mov dx, offset Labda
mov ah, 09h
int 21h

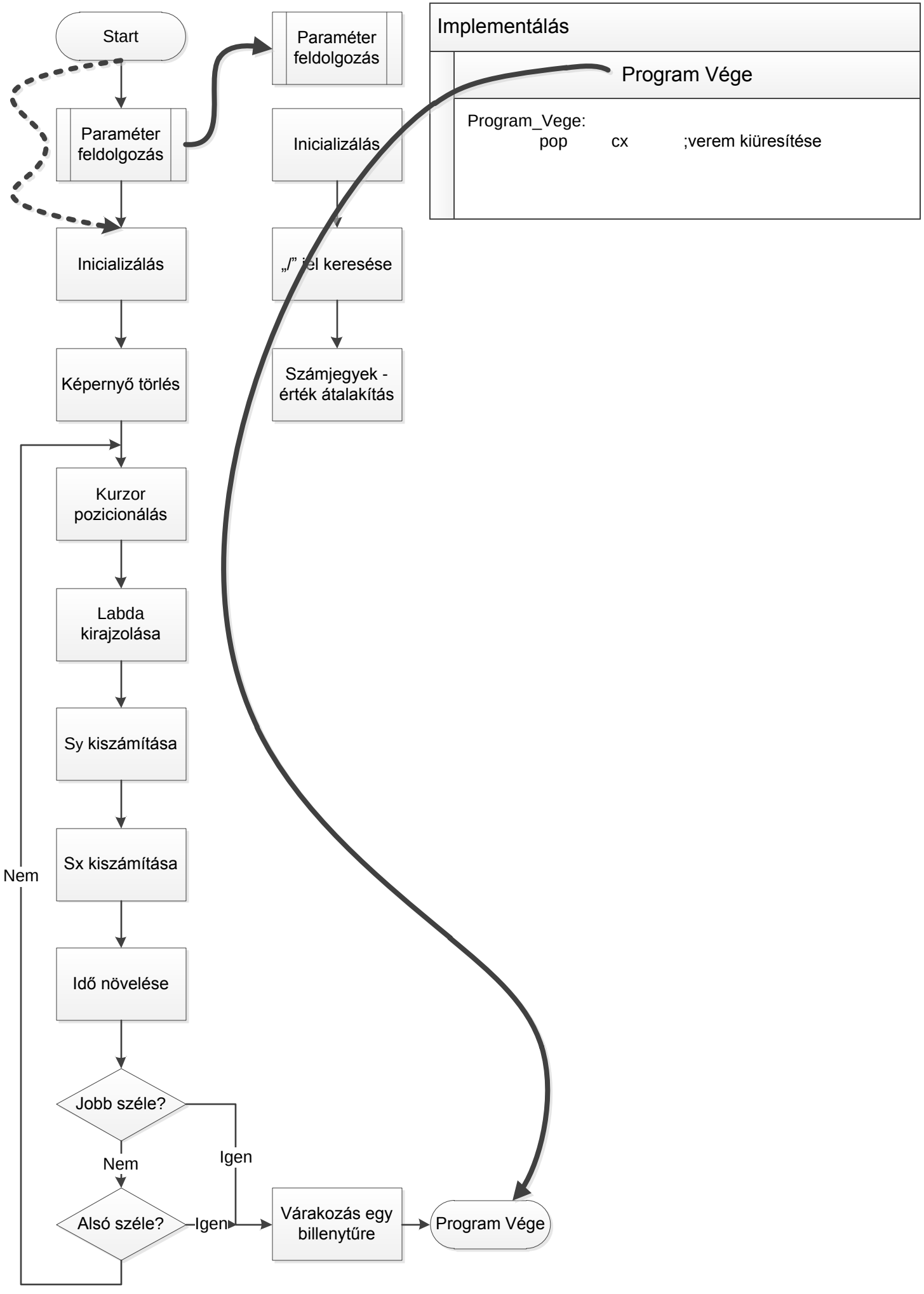

Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Képernyő törlés
Számjegyek - érték átalakítás
Kurzor pozicionálás
Kesleltet
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Nem
Idő növelése
Jobb széle?
Igen
Nem
Alsó széle?
Igen
Várakozás egy billenytűre
Program Vége
Implementálás
Sy kiszámítása
pop ax ;idő a veremben
push ax
mul al ;idő a négyzeten, eredmény ax
shr ax, 1 ;2-vel osztunk
mov di, ax

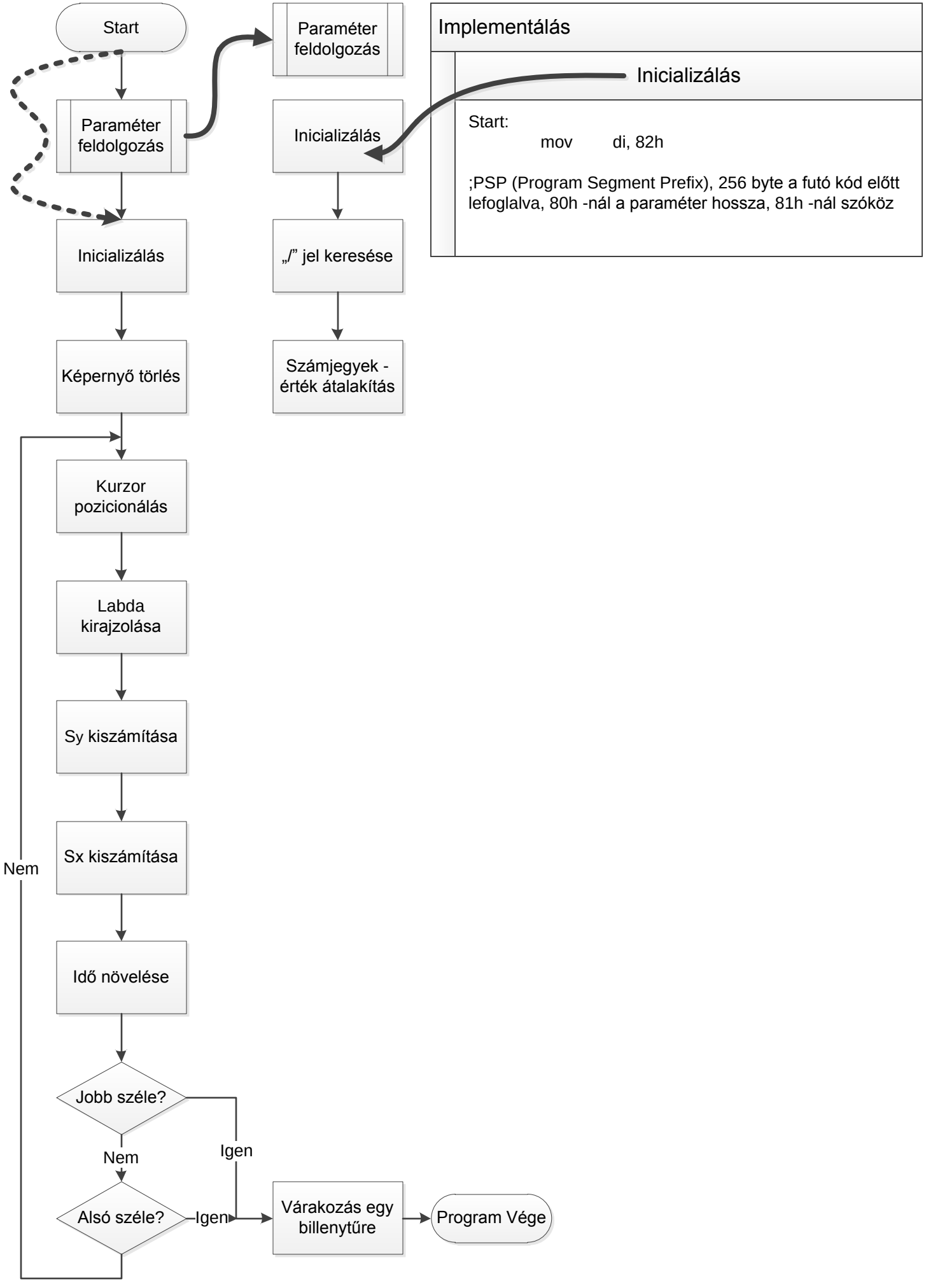

Start
Paraméter feldolgozás
Inicializálás
Képernyő törlés
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Idő növelése
Jobb széle?
Nem
Igen
Alsó széle?
Igen
Várakozás egy billenytűre
Program Vége
Paraméter feldolgozás
Inicializálás
„/” jel keresése
Számjegyek - érték átalakítás
Implementálás
Sx kiszámítása (optimális)
pop ax
inc ax
push ax
dec ax
mul cl
mov si, ax
Sx kiszámítása
pop ax
push ax
mul cl
mov si, ax
pop ax
inc ax
push ax

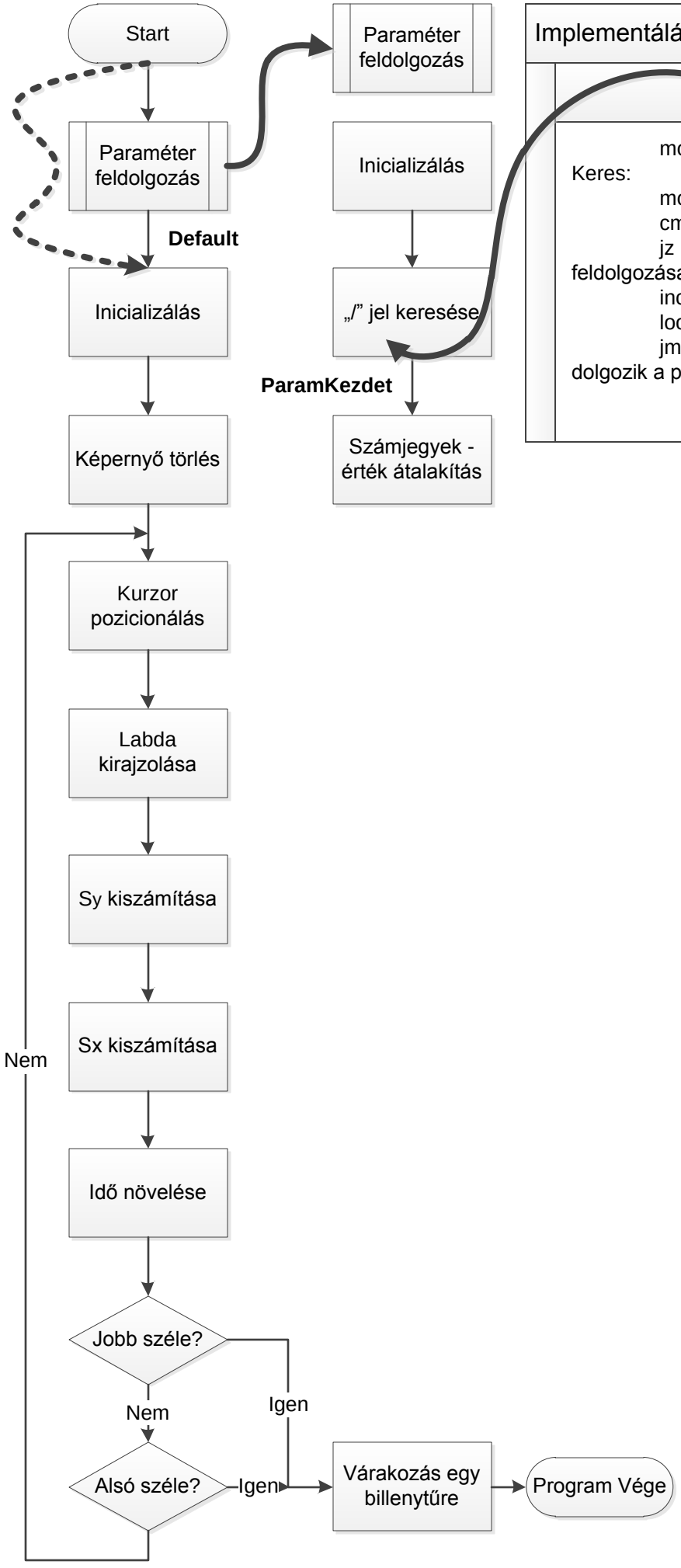

Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Képernyő törlés
Számjegyek - érték átalakítás
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Idő növelése
Jobb széle?
Nem
Igen
Alsó széle?
Igen
Var
Várakozás egy billenytűre
Program Vége
Nem
Implementálás
Jobb széle?
cmp si, 80
jnc Var


Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Képernyő törlés
Számjegyek - érték átalakítás
Rajz
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Idő növelése
Jobb széle?
Nem
Igen
Alsó széle?
Igen
Var
Várakozás egy billenytűre
Program Vége
Nem
Implementálás
Alsó széle?
cmp di, 25
jnc Var
jmp Rajz


Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Képernyő törlés
Számjegyek - érték átalakítás
Rajz
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Idő növelése
Jobb széle?
Nem
Igen
Alsó széle?
Igen
Nem
Var
Várakozás egy billenytűre
Program Vége
Implementálás
Várakozás
Var:
xor ax, ax
int 16h


Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Számjegyek - érték átalakítás
Képernyő törlés
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Idő növelése
Jobb széle?
Nem
Igen
Alsó széle?
Igen
Várakozás egy billenytűre
Program Vége
Nem
Implementálás
Program Vége
Program_Vege:
pop cx ;verem kiüresítése

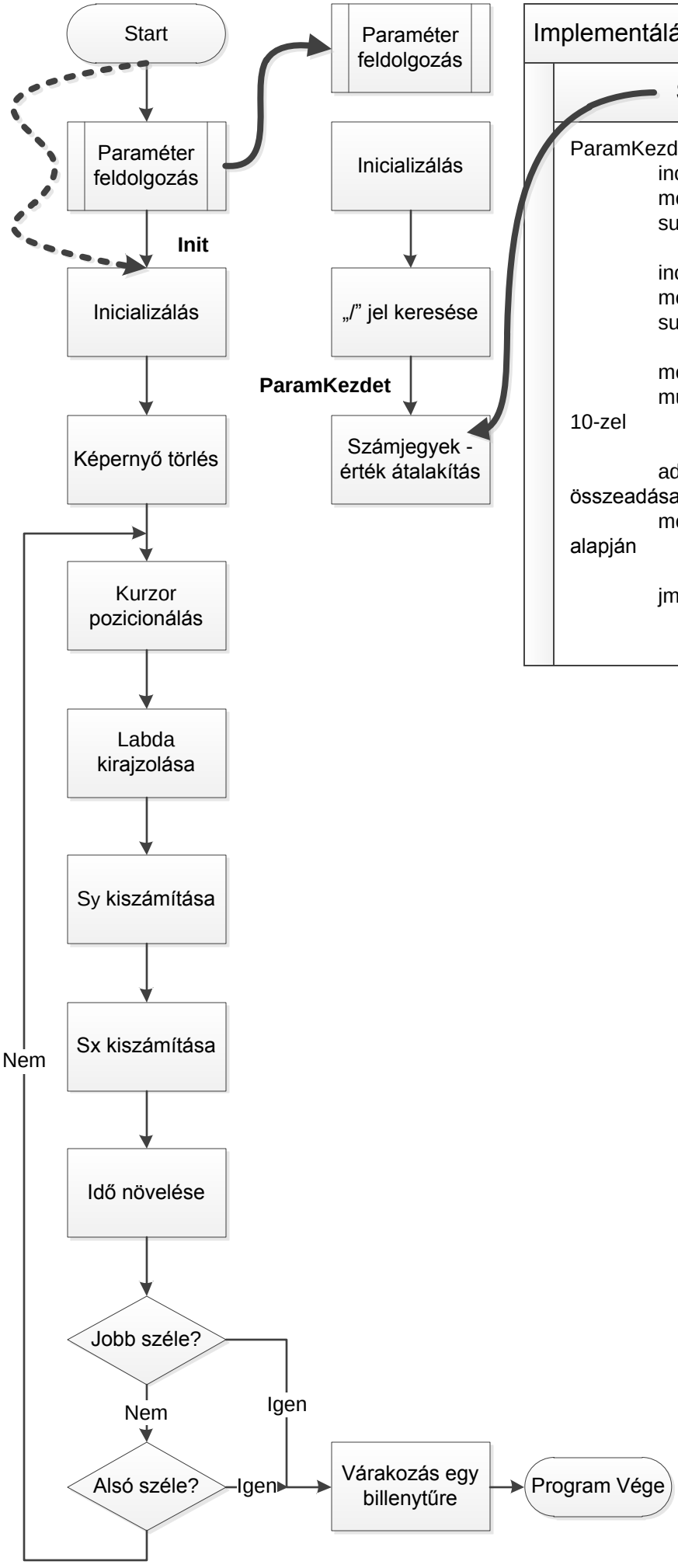

Start
Paraméter feldolgozás
Paraméter feldolgozás
Inicializálás
Inicializálás
„/” jel keresése
Számjegyek - érték átalakítás
Képernyő törlés
Kurzor pozicionálás
Labda kirajzolása
Sy kiszámítása
Sx kiszámítása
Idő növelése
Jobb széle?
Nem
Igen
Alsó széle?
Igen
Várakozás egy billenytűre
Program Vége
Nem
Implementálás
Inicializálás
Start:
mov di, 82h
;PSP (Program Segment Prefix), 256 byte a futó kód előtt lefoglalva, 80h -nál a paraméter hossza, 81h -nál szóköz



Implementálás

"/" jel keresése

```
        mov     cx, 10      ;ciklus változó beállítása
Keres:
        mov     dl, [di]
        cmp     dl, "/"
        jz      ParamKezdet     ;talált, számjegyek
feldolgozása
        inc     di
        loop    Keres
        jmp     Default     ;nem talált, default értékkel
dolgozik a program a továbbiakban
```



Implementálás

Számjegyek - érték átalakítás

```
ParamKezdet:
        inc     di
        mov     bl, [di]    ;első számjegy, tízesek
        sub     bl, 48

        inc     di
        mov     bh, [di]    ;második számjegy, egyesek
        sub     bh, 48

        mov     ax, 10
        mul     bl          ;tízes számjegy megszorzása
10-zel

        add     al, bh      ;tízesek és egyesek
összeadása
        mov     cx, ax      ;v0 beállítása a paraméter
alapján

        jmp     Init
```