Fealadat3: labda.asm

Feladat meghatározása

A program célja az assembly rutinok időzítési lehetőségeinek bemutatása. Az időzítés az AH00, INT1Ah funkció segítségével történik. A program egy „labda” leesését szimulálja. A labdát egy „O” karakter szimbolizálja, amely fokozatosan gyorsulva esik le, majd visszapattan. A visszapattanás veszteségmentes, így a labda fokozatosan lassulva eléri a kiindulási pozíciót.

Program bemutatása:

- A programot először a késleltető rutinok nélkül kell megírni. A billentyű figyelés helyett az AH00, INT16h funkcióval megvalósítható billentyű leütésre történő várakozást kell megírni. Így a program csak akkor lép a következő iterációra, ha lenyomunk egy billentyűt. Ellenőrizni kell, hogy a „labda” nem megy ki a képből. A legalsó pozíciót elérve felfelé mozog tovább, a legfelső pozíció elérését követően lefele halad.

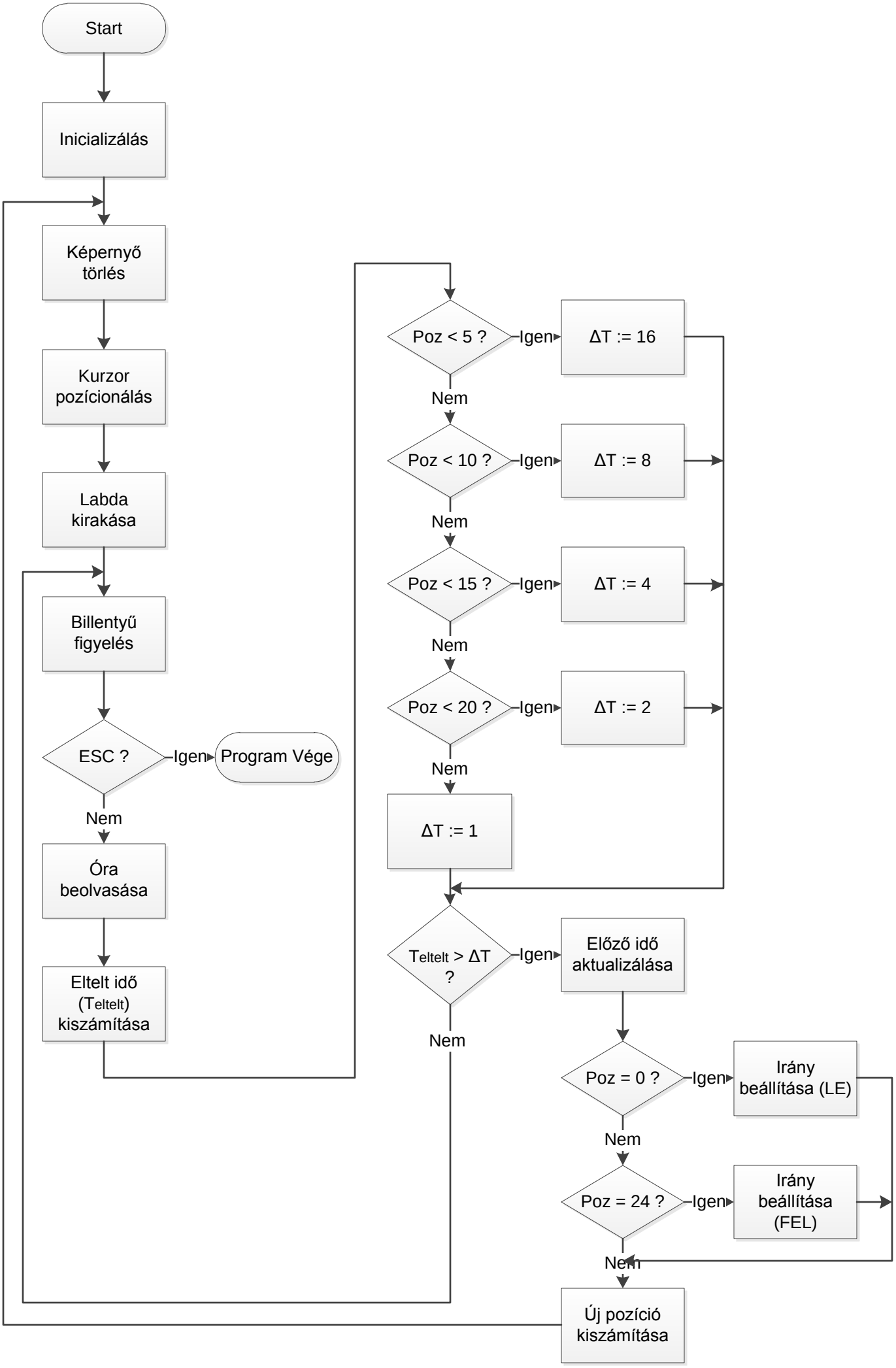
- A letesztelt programban módosítani kell a billentyű leütésre váró részt az AH01, INT16h funkcióra. Ez a funkció nem állítja meg a program futását, mindössze a FLAG-ek állapotát, illetve az AL értékét módosítja, ha volt billentyű leütés. Be kell fejezni a késleltető programrészt. A késleltetés alapeleme az AH00, INT1Ah funkció, mely a rendszeróra CX:DX regiszterpárba. Egy számláló-változás kb. 1/18 sec.

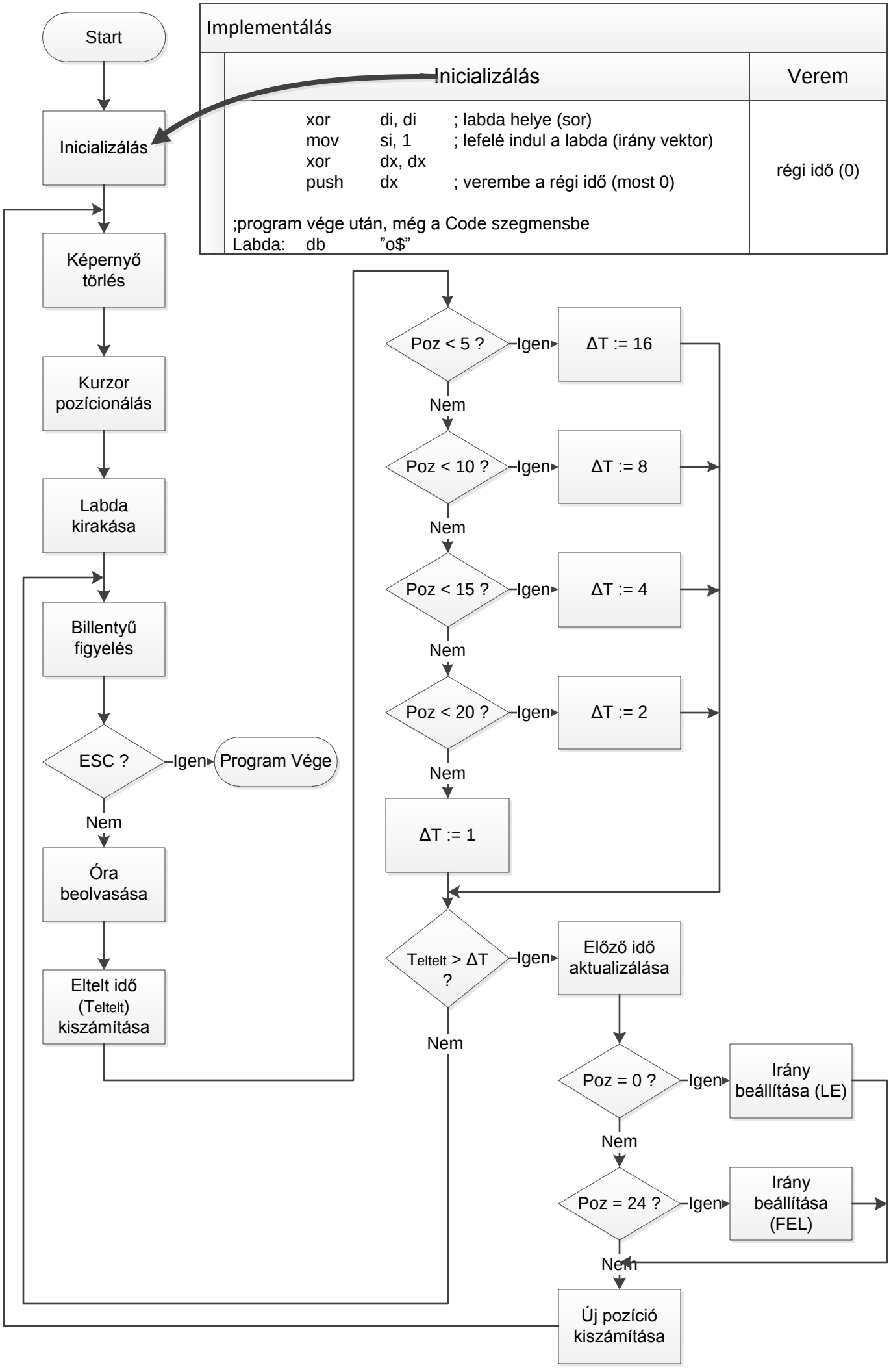
Implementálás

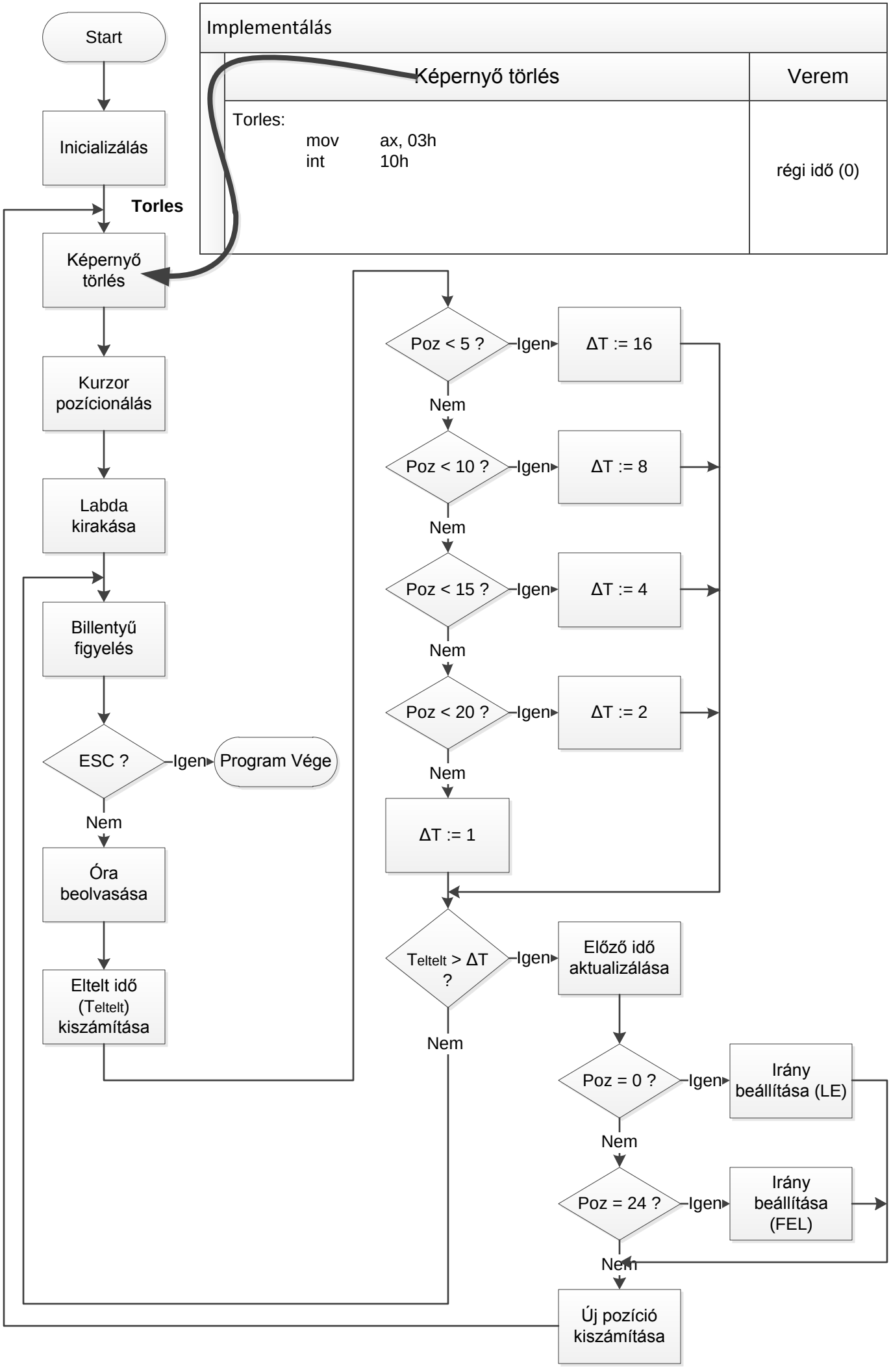
```
Code    Segment
        assume CS:Code,
        DS:Data, SS:Stack
Start:
        mov     ax, Code
        mov     DS, ax

        xor     di, di
        mov     si, 1

        xor     dx, dx
        push    dx
Torles:
        mov     ax, 03h
        int     10h

        mov     dx, di
        mov     dh, dl

        mov     dl, 40
        xor     bh, bh
        mov     ah, 02h
        int     10h

        mov     dx, offset Labda
        mov     ah, 09h
        int     21h
Kesleltet:
        mov     ah, 01h
        int     16h
        ;jnz    Program_Vege

        jz      nincsbill
        mov     ah, 00h
        int     16h
        cmp     al, 27
        jz      Program_Vege
nincsbill:
        xor     ah, ah
        int     1ah

        pop     cx
        push    cx
        mov     ax, dx
        sub     dx, cx
        push    ax

        cmp     di, 5
        jnc     Ido1
        mov     al, 16
        jmp     Beallit
Ido1:
        cmp     di, 10
        jnc     Ido2
        mov     al, 8
        jmp     Beallit
Ido2:
        cmp     di, 15
        jnc     Ido3
        mov     al, 4
        jmp     Beallit
Ido3:
```

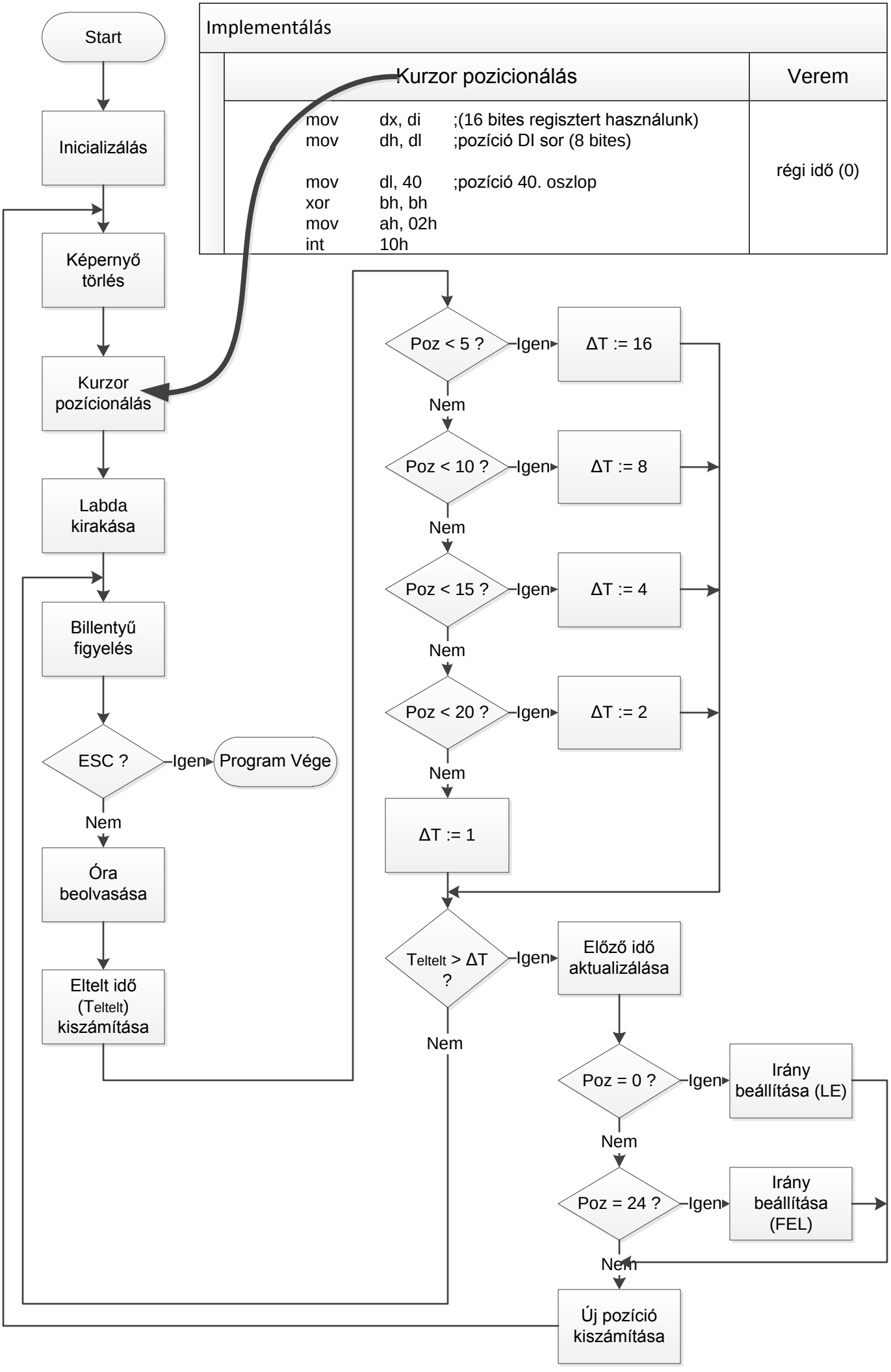
Implementálás

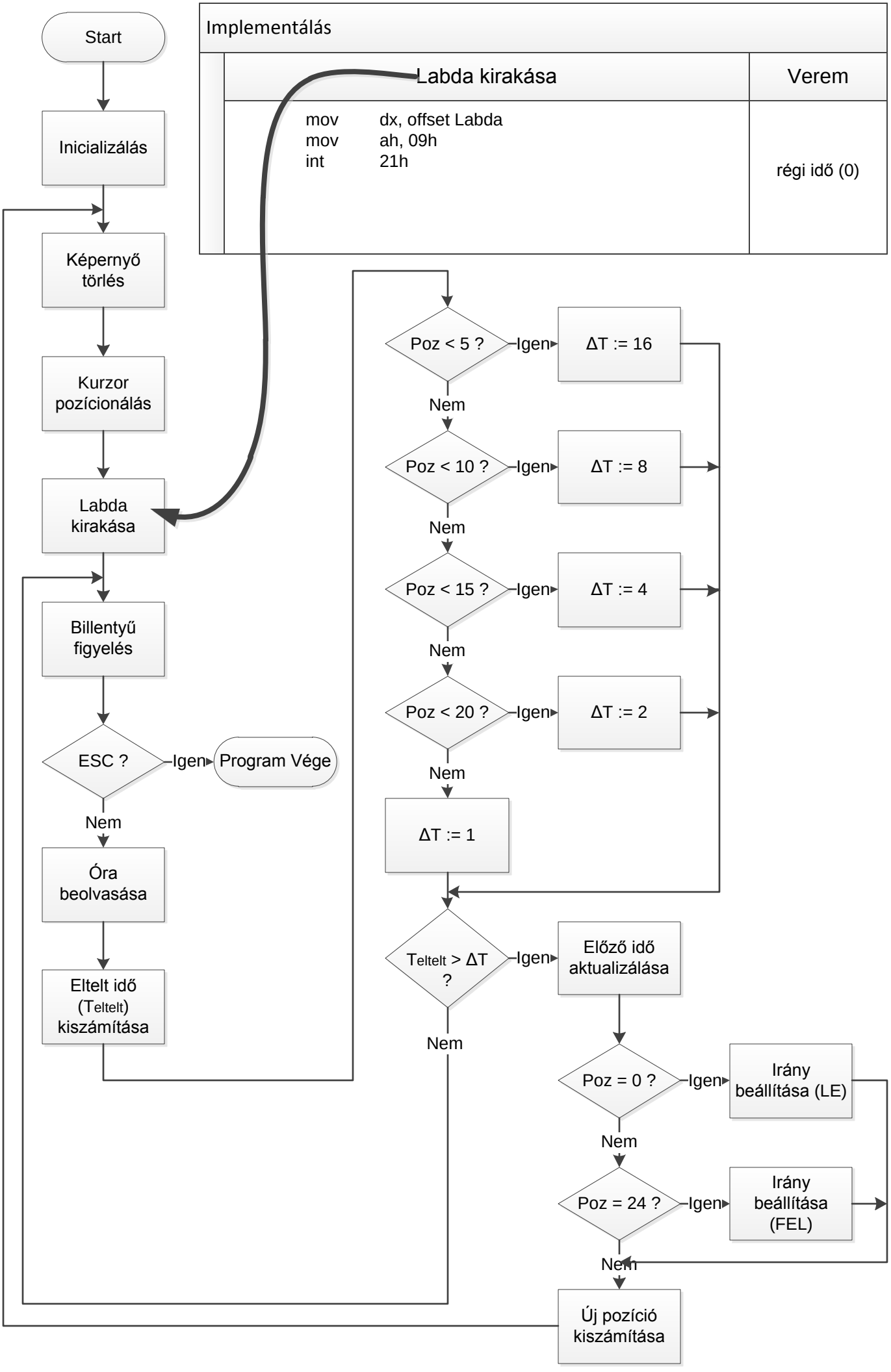
```
        cmp     di, 20
        jnc     Ido4
        mov     al, 2
        jmp     Beallit
Ido4:
        mov     al, 1
Beallit:
        xor     ah, ah
        cmp     dx, ax

        pop     ax

        jc      Kesleltet

        pop     cx
        push    ax

        cmp     di, 0
        jz      Lefele

        cmp     di, 24
        jz      Felfele
Mozgas:
        add     di, si
        jmp     Torles
Lefele:
        mov     si, 1
        jmp     Mozgas
Felfele:
        mov     si, -1
        jmp     Mozgas

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

Labda:
        db      "O$"

Code    Ends

Data    Segment
Data    Ends

Stack   Segment
Stack   Ends

        End Start
```

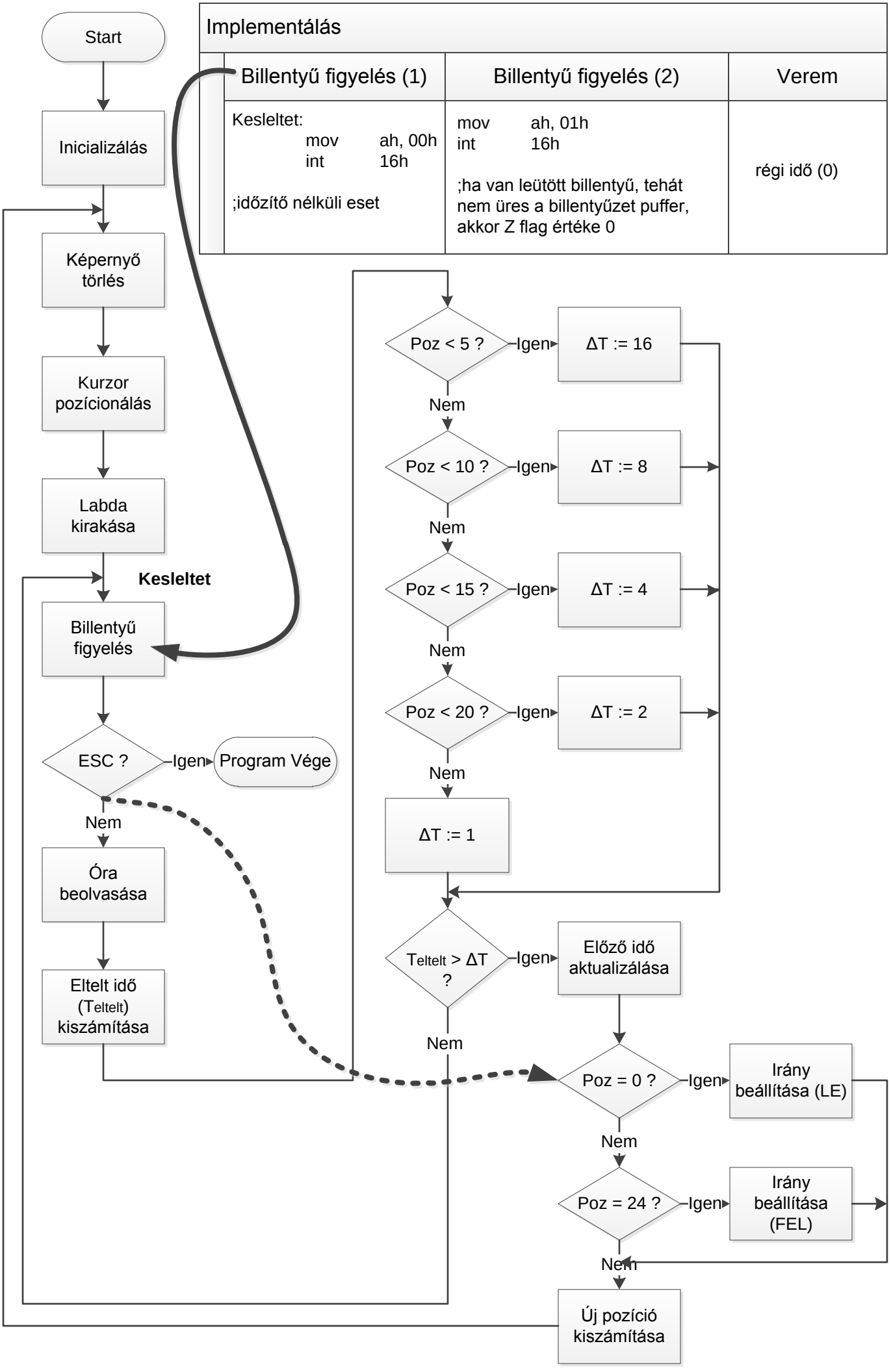

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

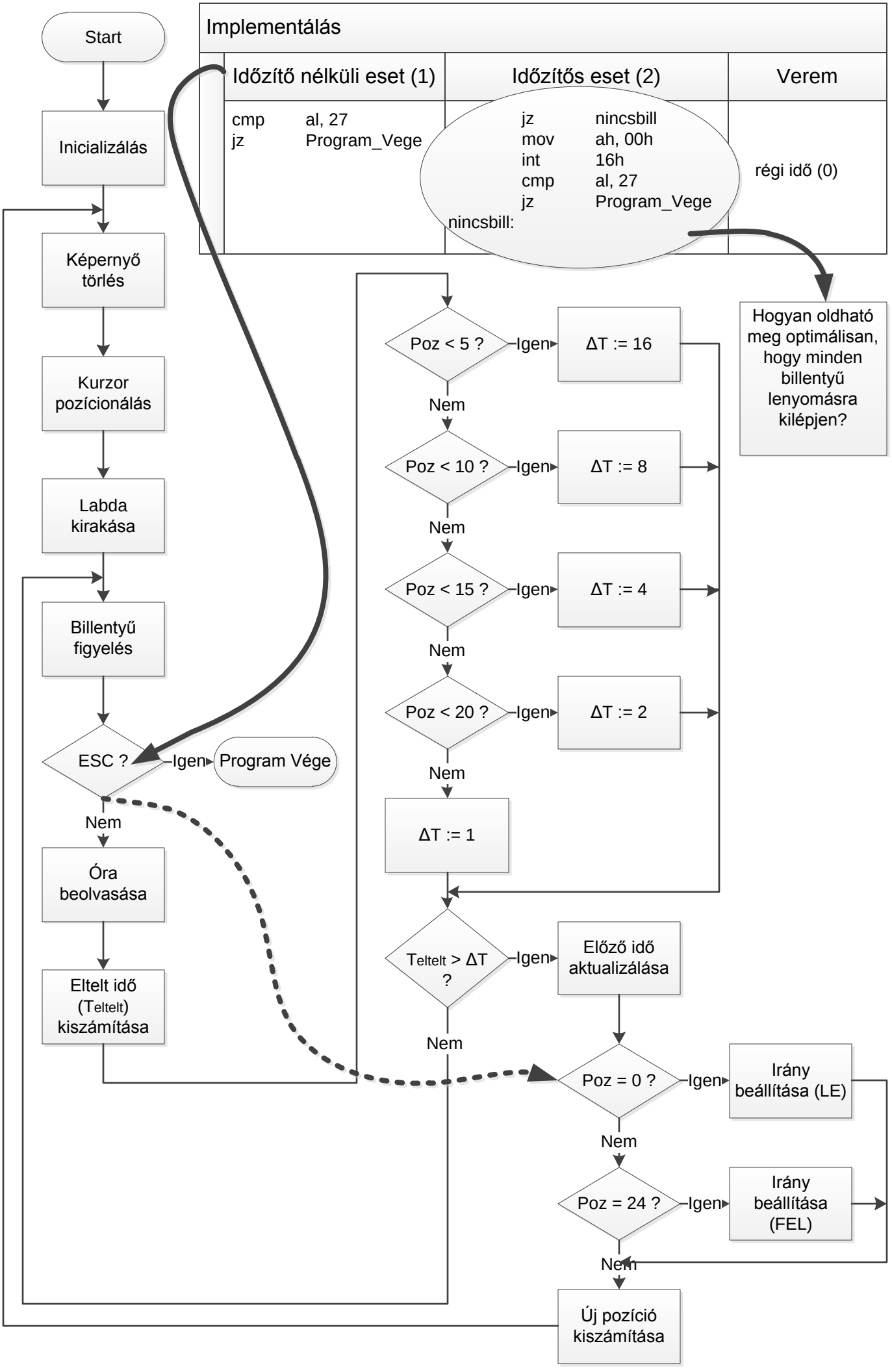

Start
Implementálás
Inicializálás
Verem
xor di, di ; labda helye (sor)
mov si, 1 ; lefelé indul a labda (irány vektor)
xor dx, dx
push dx ; verembe a régi idő (most 0)
régi idő (0)
;program vége után, még a Code szegmensbe
Labda: db ”o$”
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

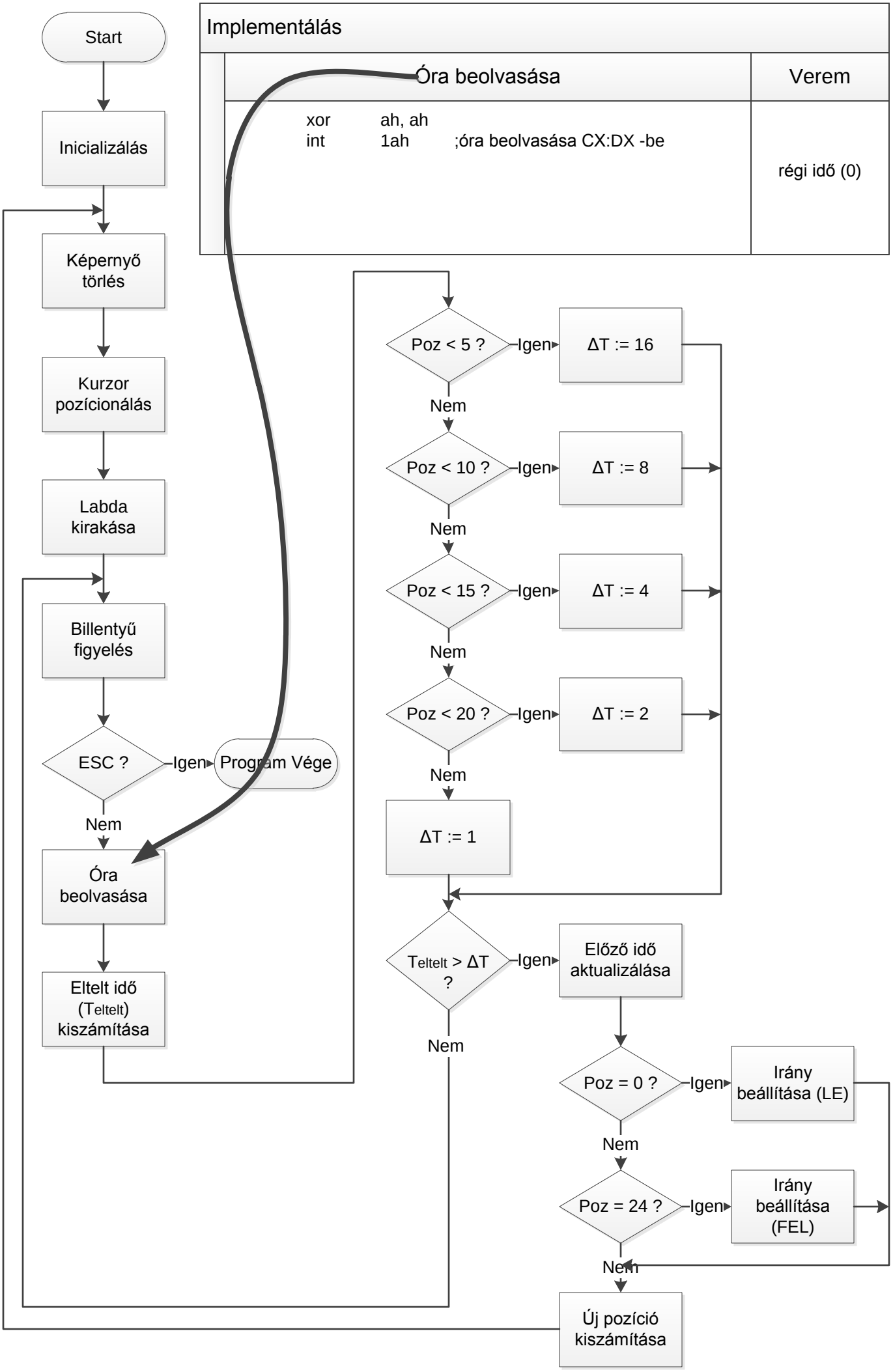

Start
Inicializálás
Torles
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Képernyő törlés
Verem
Torles:
mov ax, 03h
int 10h
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

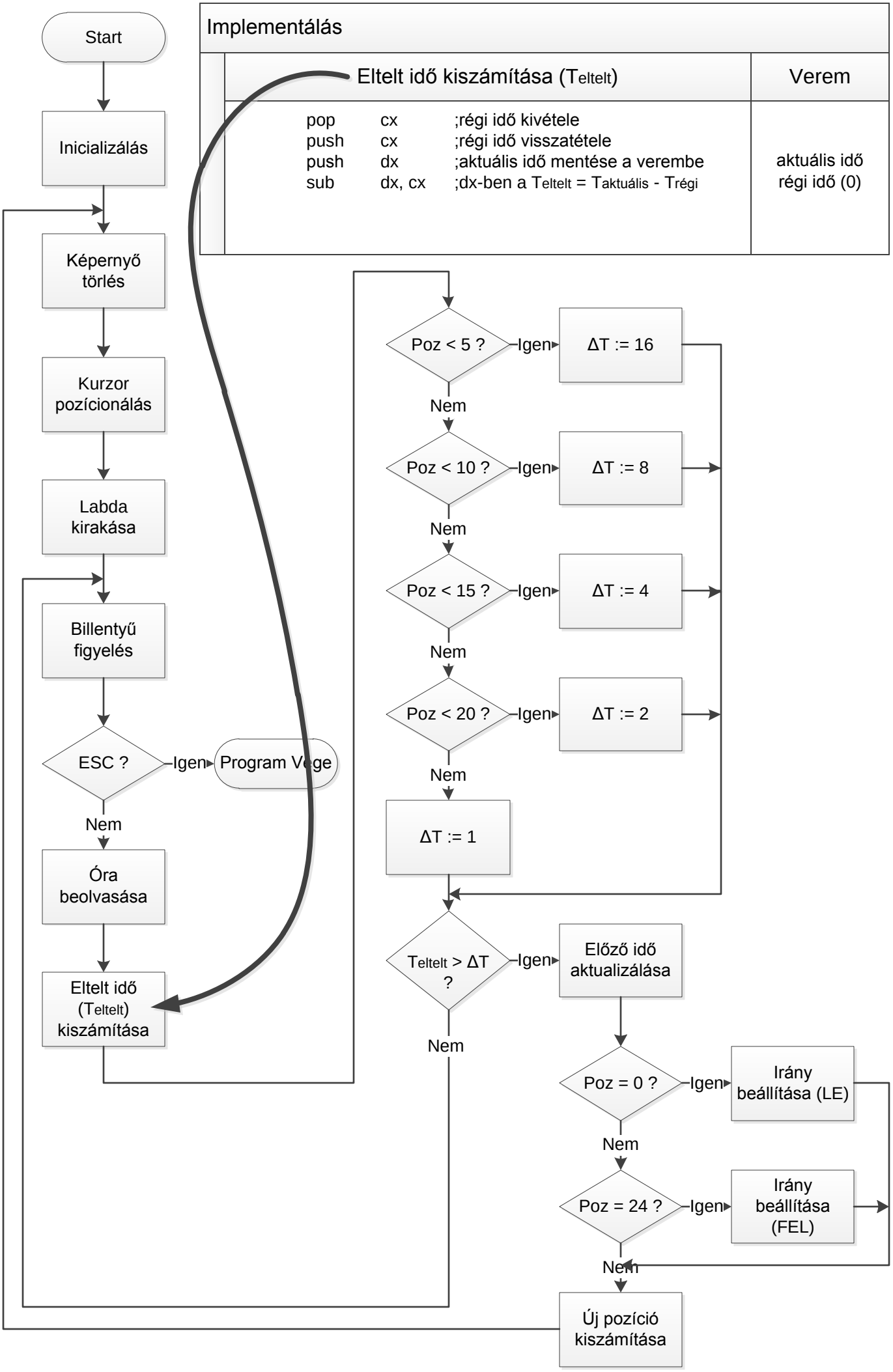

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Kurzor pozicionálás
Verem
mov dx, di ;(16 bites regisztert használunk)
mov dh, dl ;pozíció DI sor (8 bites)
mov dl, 40 ;pozíció 40. oszlop
xor bh, bh
mov ah, 02h
int 10h
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

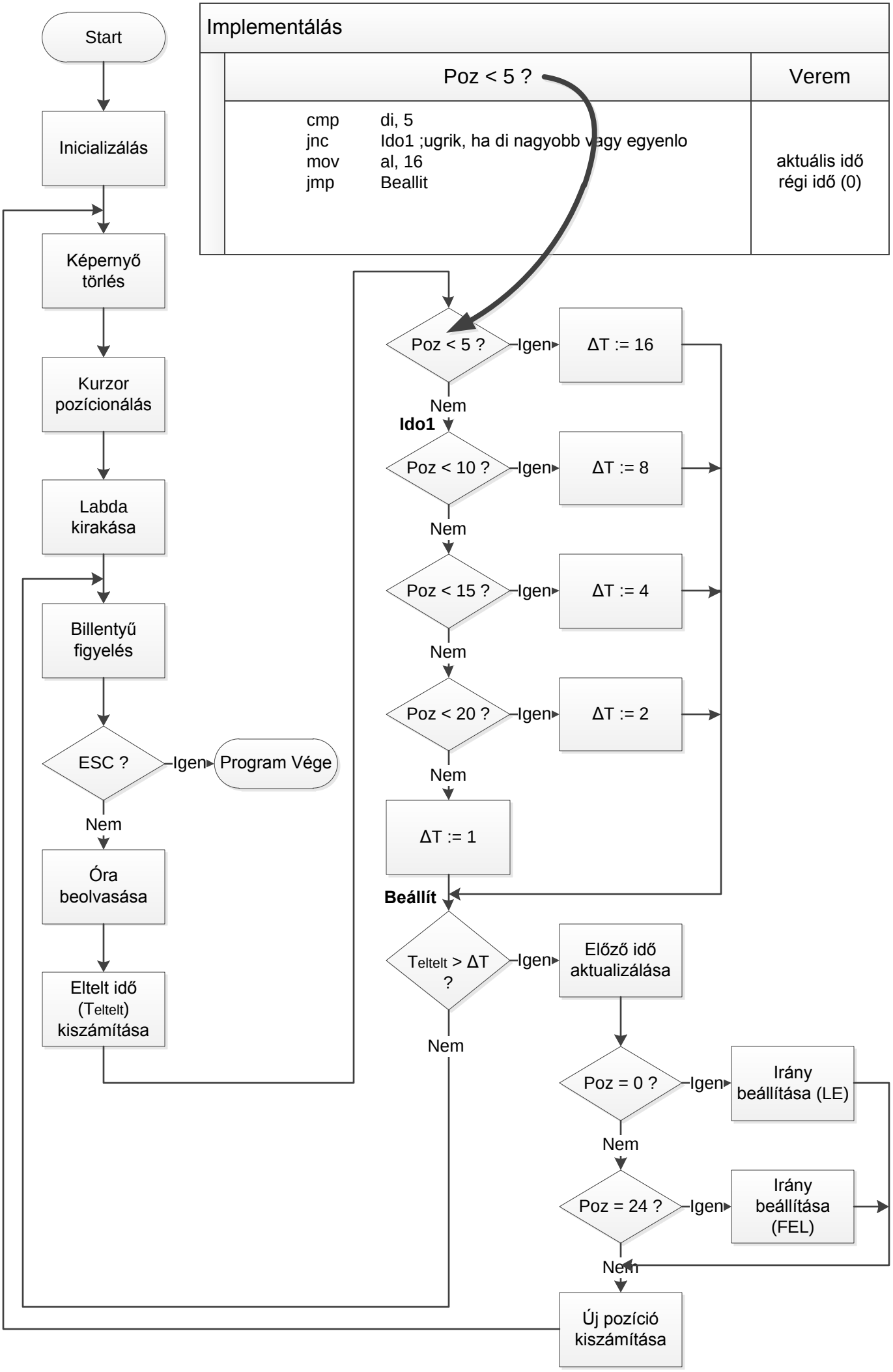

Start
Implementálás
Labda kirakása
Verem
mov dx, offset Labda
mov ah, 09h
int 21h
régi idő (0)
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

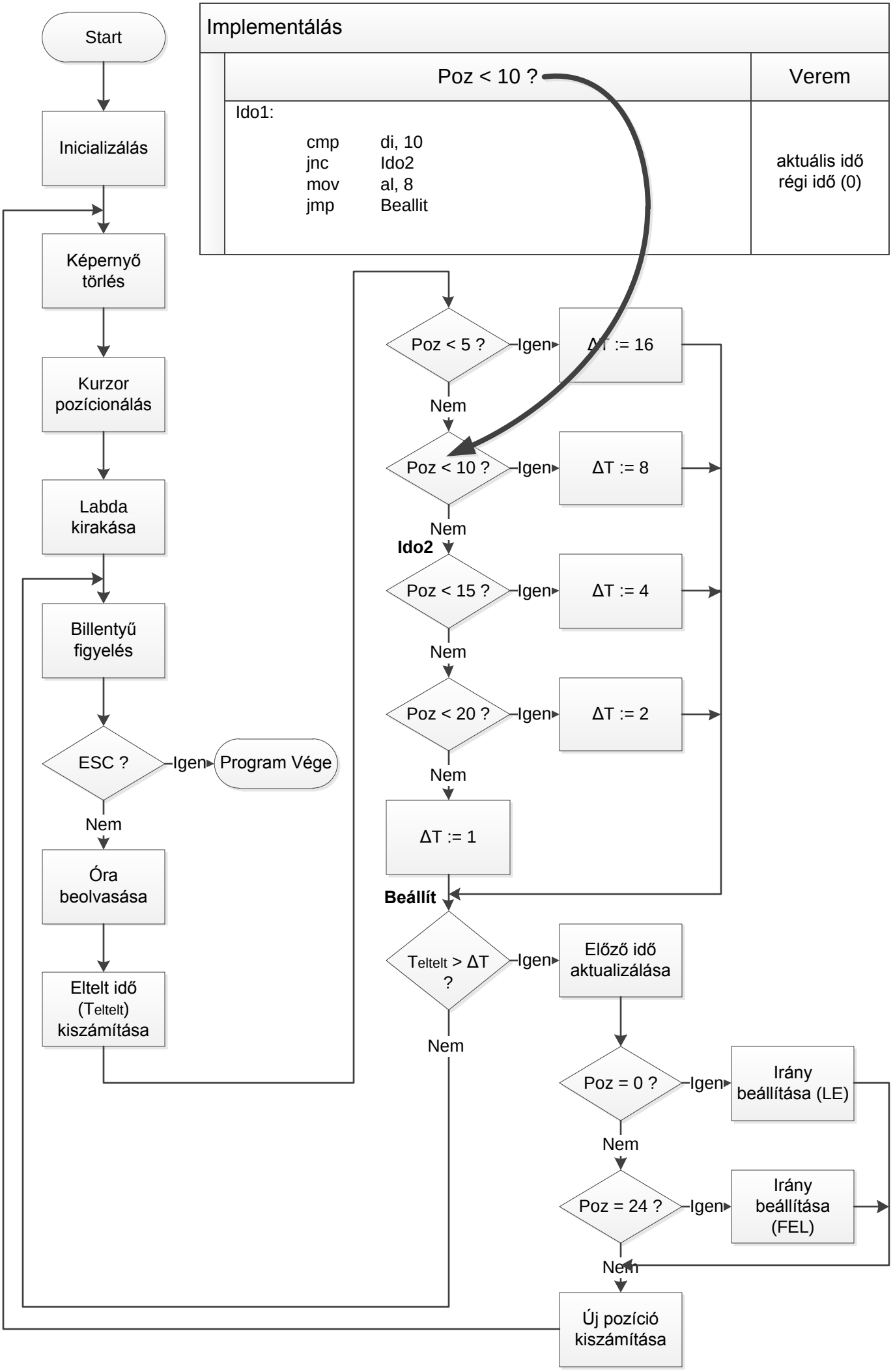

Implementálás
Billentyű figyelés (1)
Billentyű figyelés (2)
Verem
Kesleltet:
mov ah, 00h
int 16h
;időzítő nélküli eset
mov ah, 01h
int 16h
;ha van leütött billentyű, tehát nem üres a billentyűzet puffer, akkor Z flag értéke 0
régi idő (0)
Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Kesleltet
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

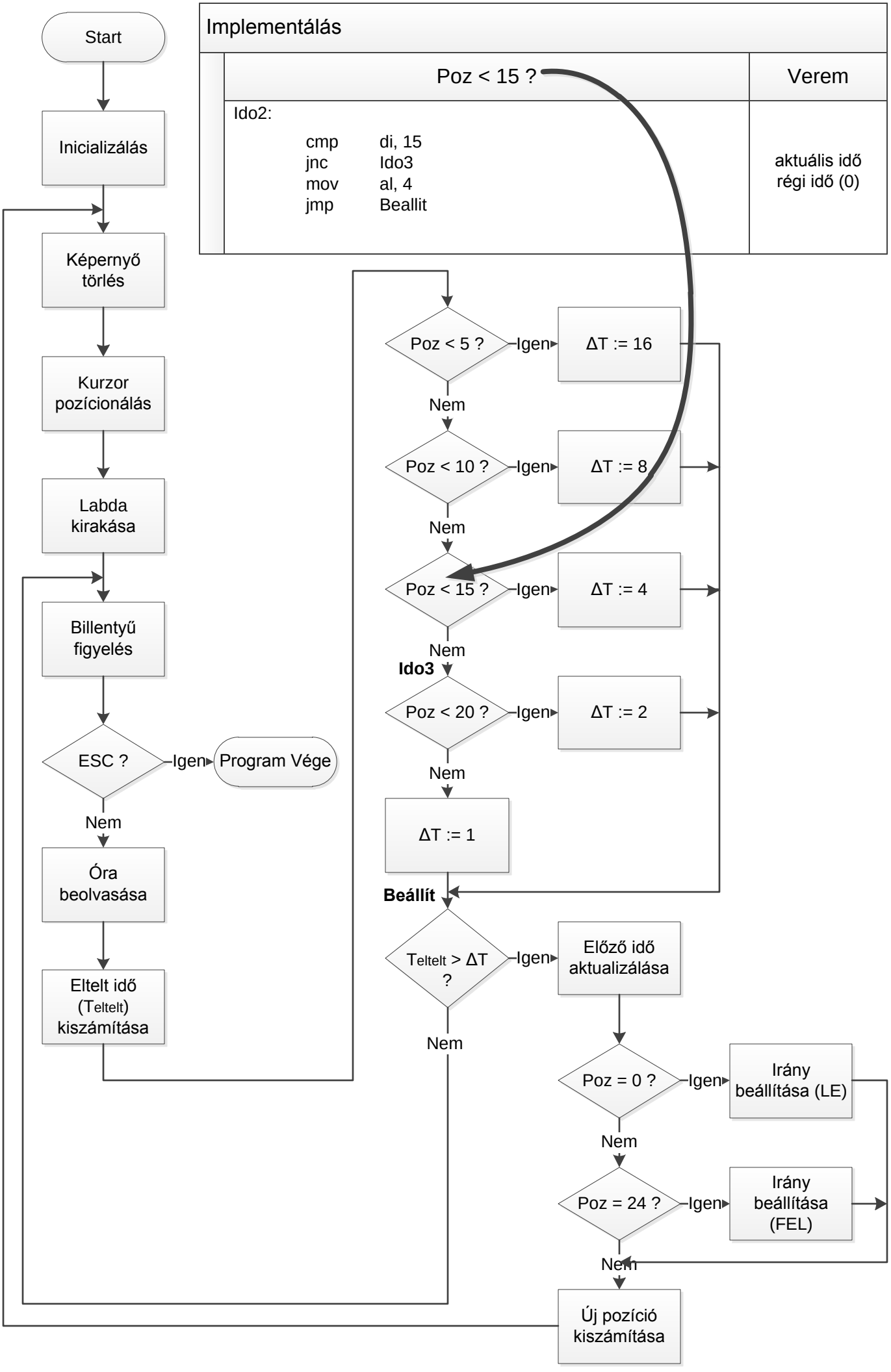

Start
Implementálás
Időzítő nélküli eset (1)
Időzítős eset (2)
Verem
cmp al, 27
jz Program_Vege
jz nincsbill
mov ah, 00h
int 16h
cmp al, 27
jz Program_Vege
nincsbill:
régi idő (0)
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása
Hogyan oldható meg optimálisan, hogy minden billentyű lenyomásra kilépjen?

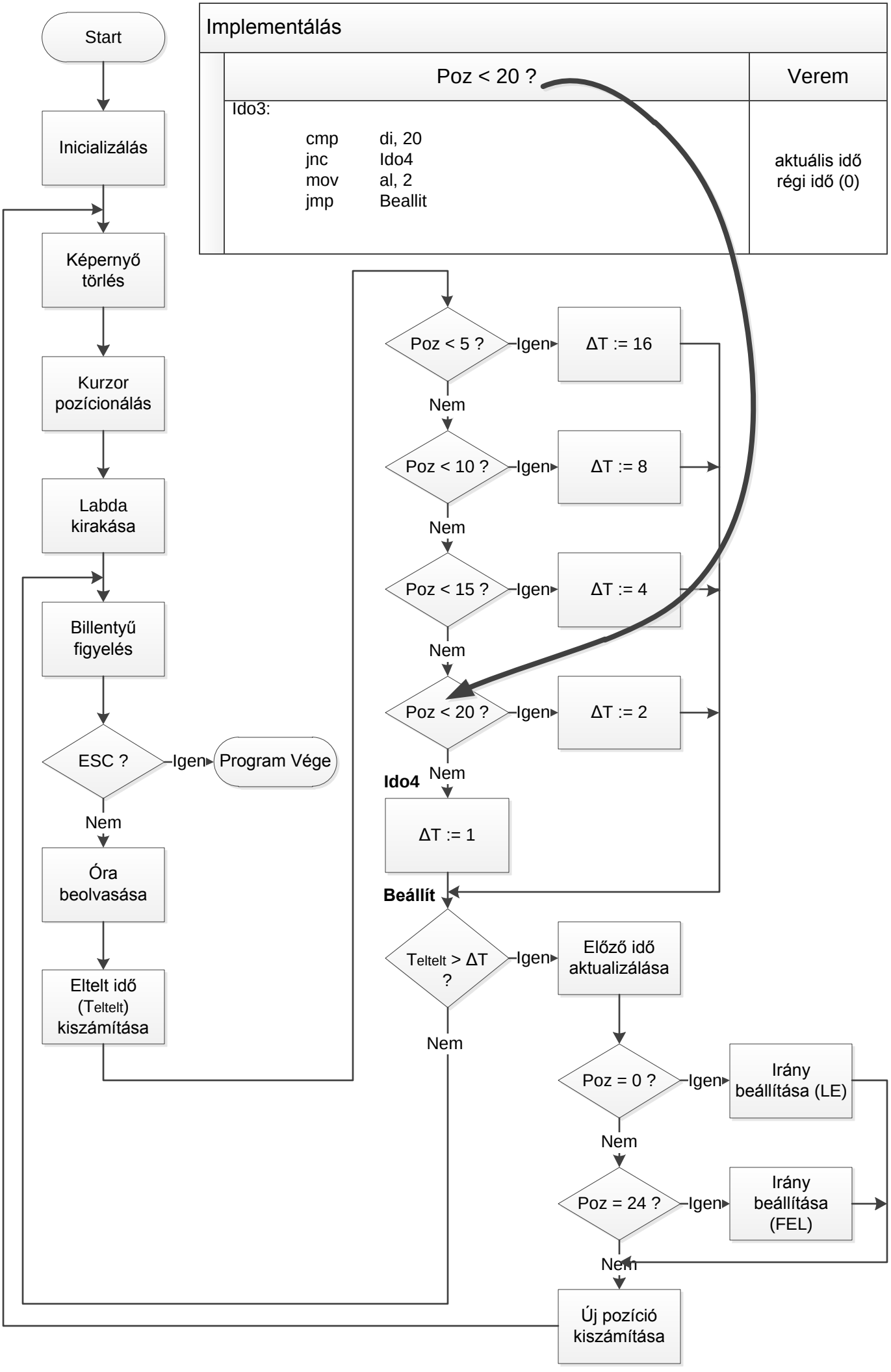

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Óra beolvasása
Verem
xor ah, ah
int 1ah ;óra beolvasása CX:DX -be
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

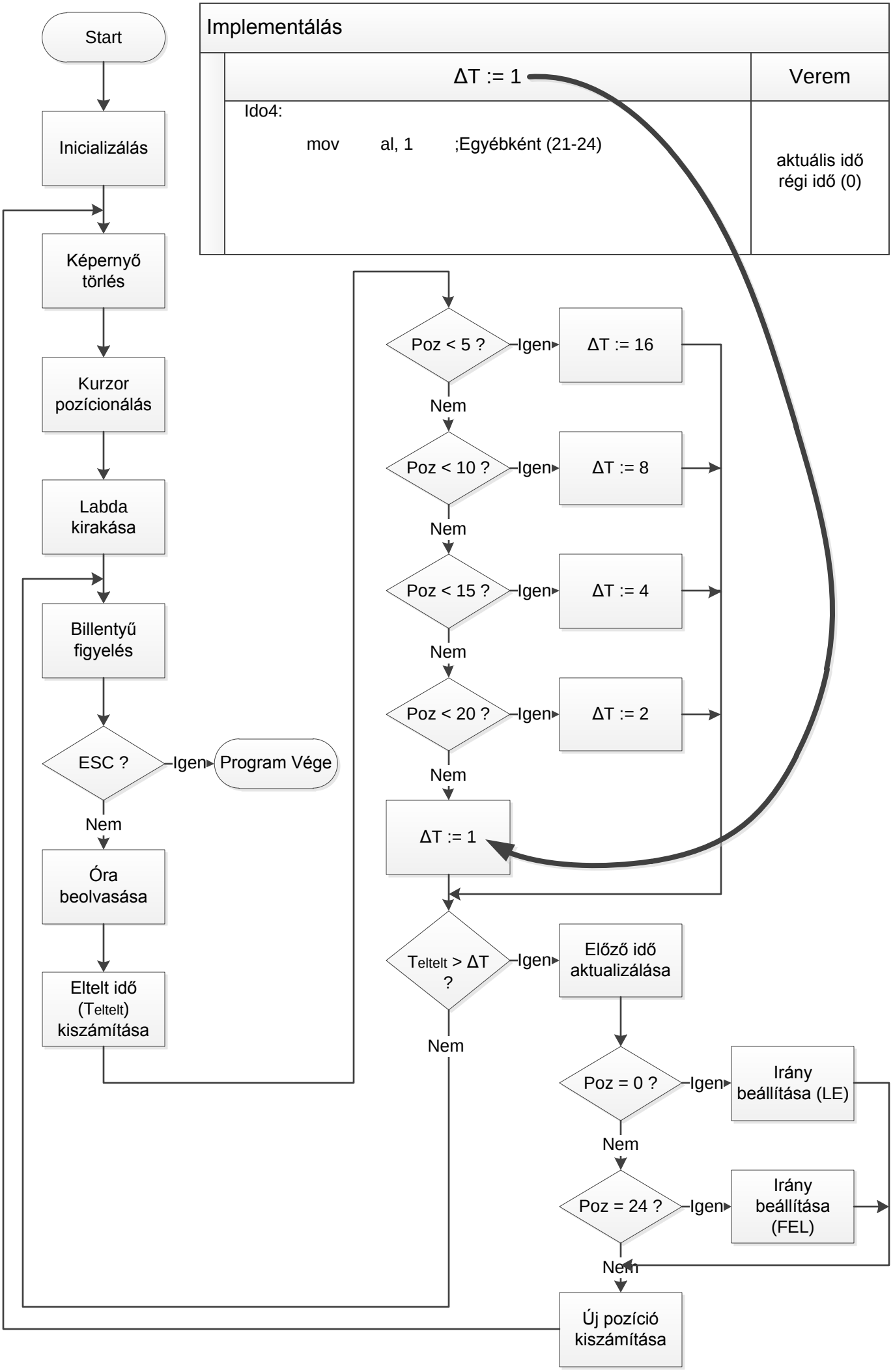

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Eltelt idő kiszámítása (Teltelt)
Verem
pop cx ;régi idő kivétele
push cx ;régi idő visszatétele
push dx ;aktuális idő mentése a verembe
sub dx, cx ;dx-ben a Teltelt = Taktuális - Trégi
aktuális idő
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

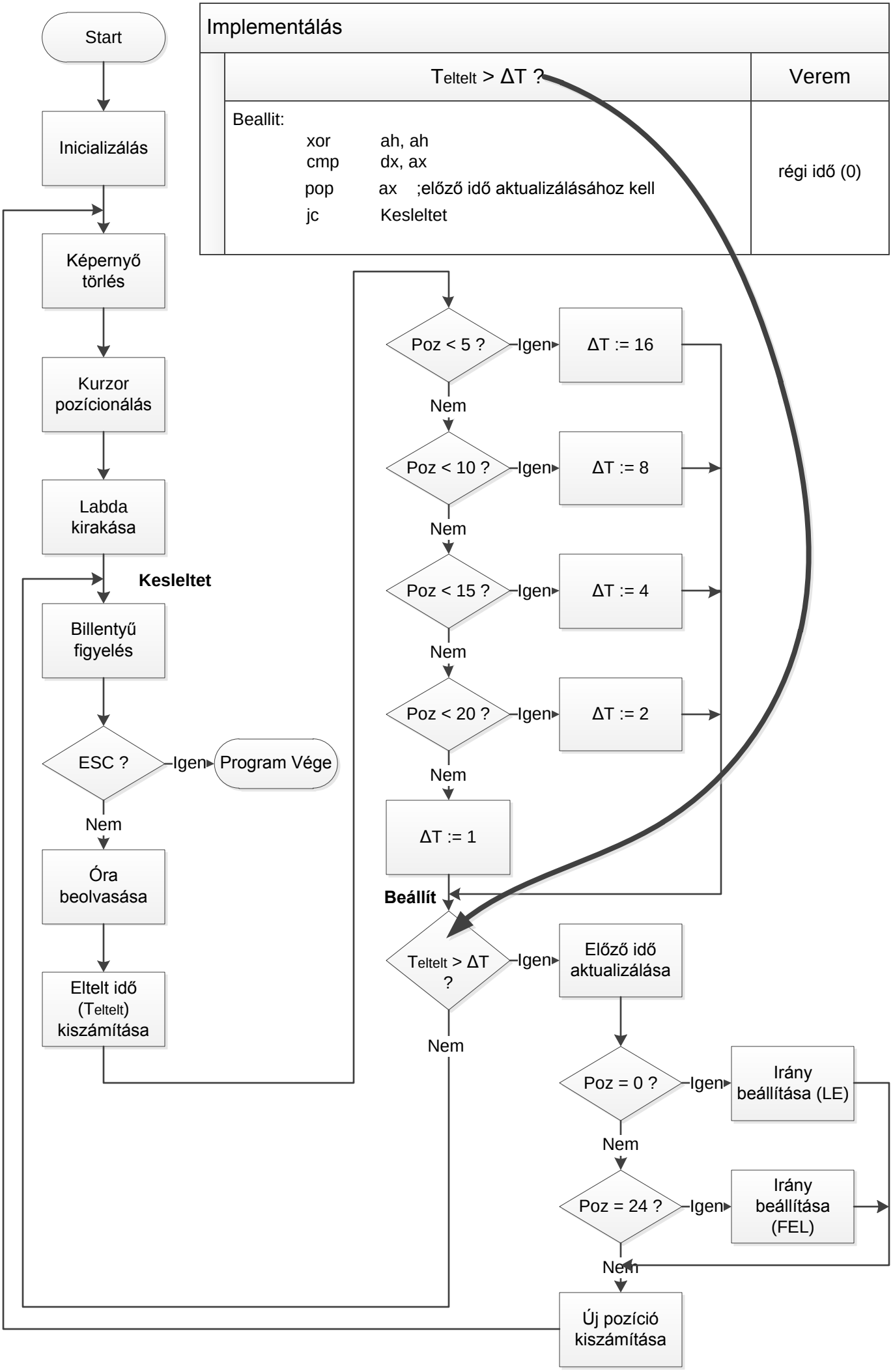

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Poz < 5 ?
Verem
cmp di, 5
jnc Ido1 ;ugrik, ha di nagyobb vagy egyenlo
mov al, 16
jmp Beallit
aktuális idő
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Ido1
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Beállít
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

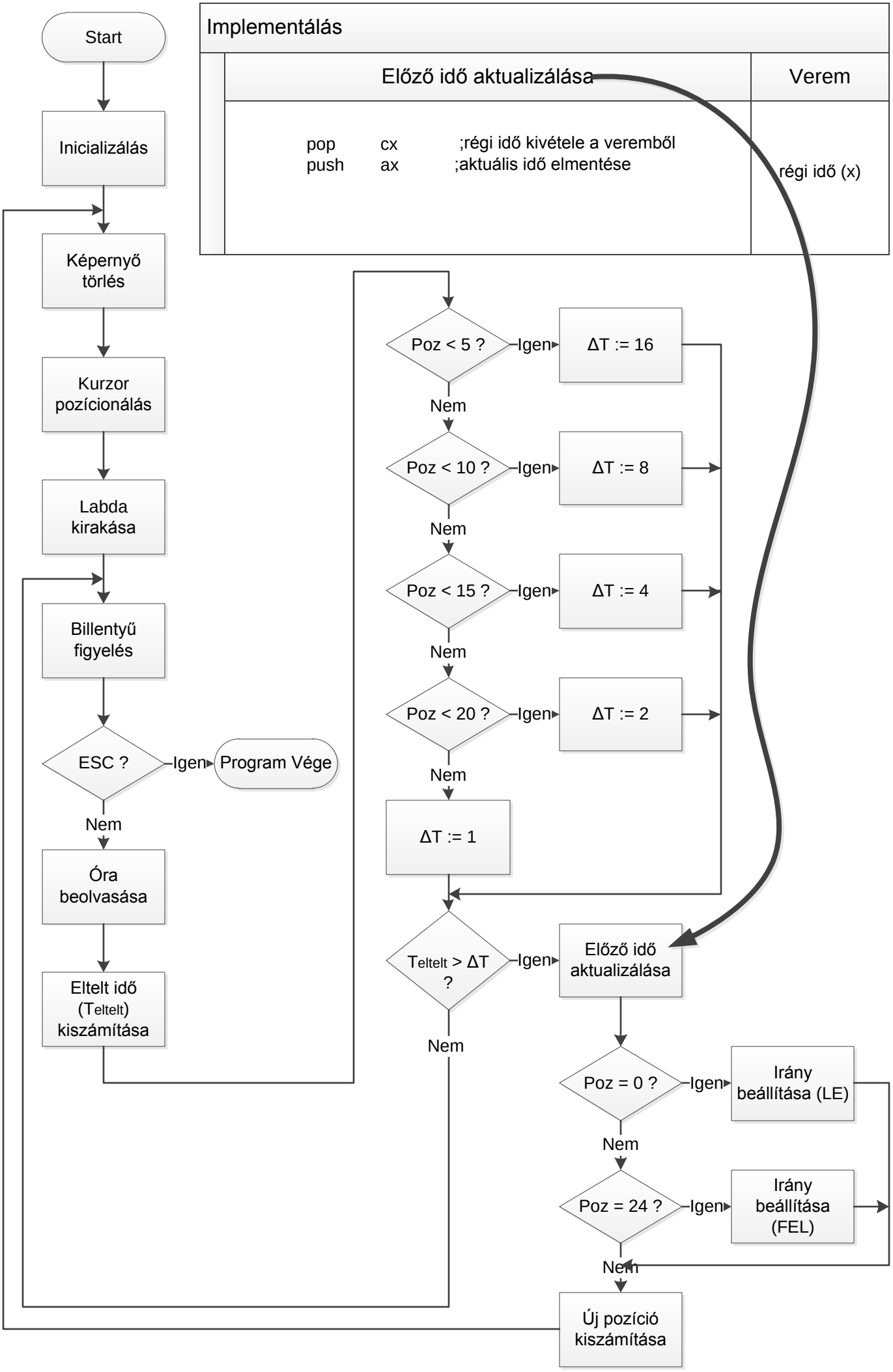

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Poz < 10 ?
Verem
Ido1:
cmp di, 10
jnc Ido2
mov al, 8
jmp Beallit
aktuális idő
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Ido2
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Beállít
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

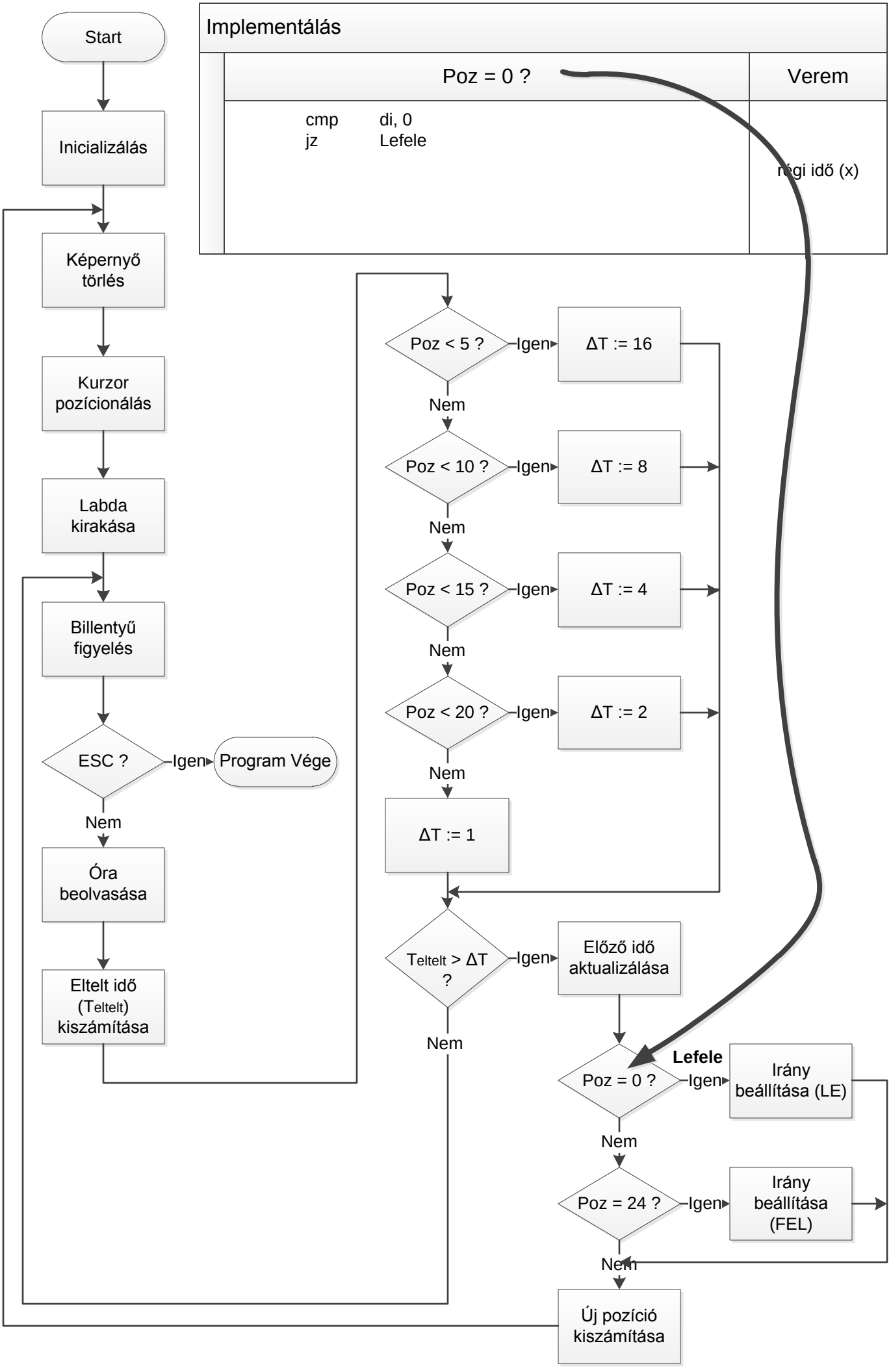

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Poz < 15 ?
Verem
Ido2:
cmp di, 15
jnc Ido3
mov al, 4
jmp Beallit
aktuális idő
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Ido3
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Beállít
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

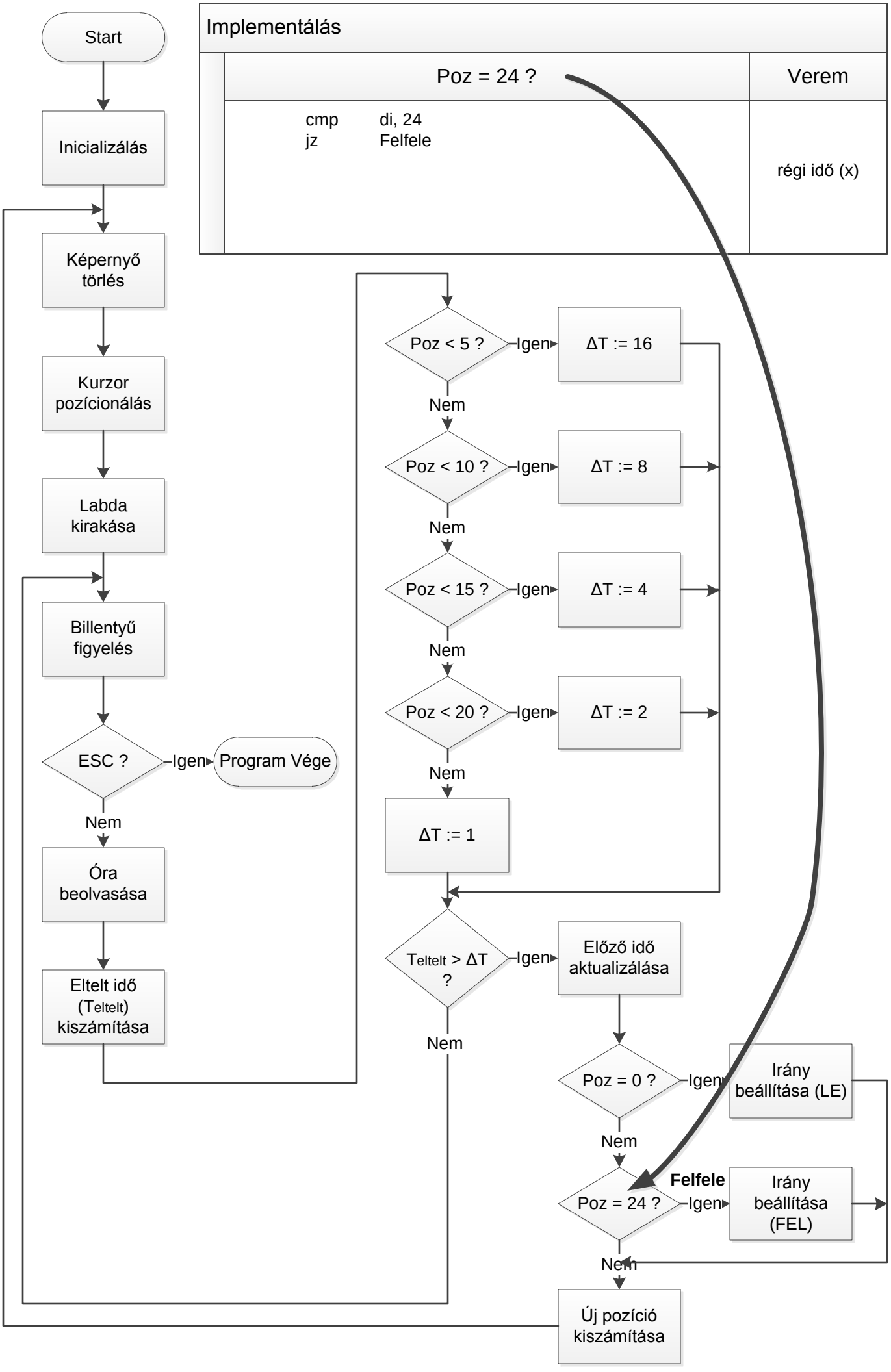

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Poz < 20 ?
Verem
Ido3:
cmp di, 20
jnc Ido4
mov al, 2
jmp Beallit
aktuális idő
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
Ido4
ΔT := 1
Beállít
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

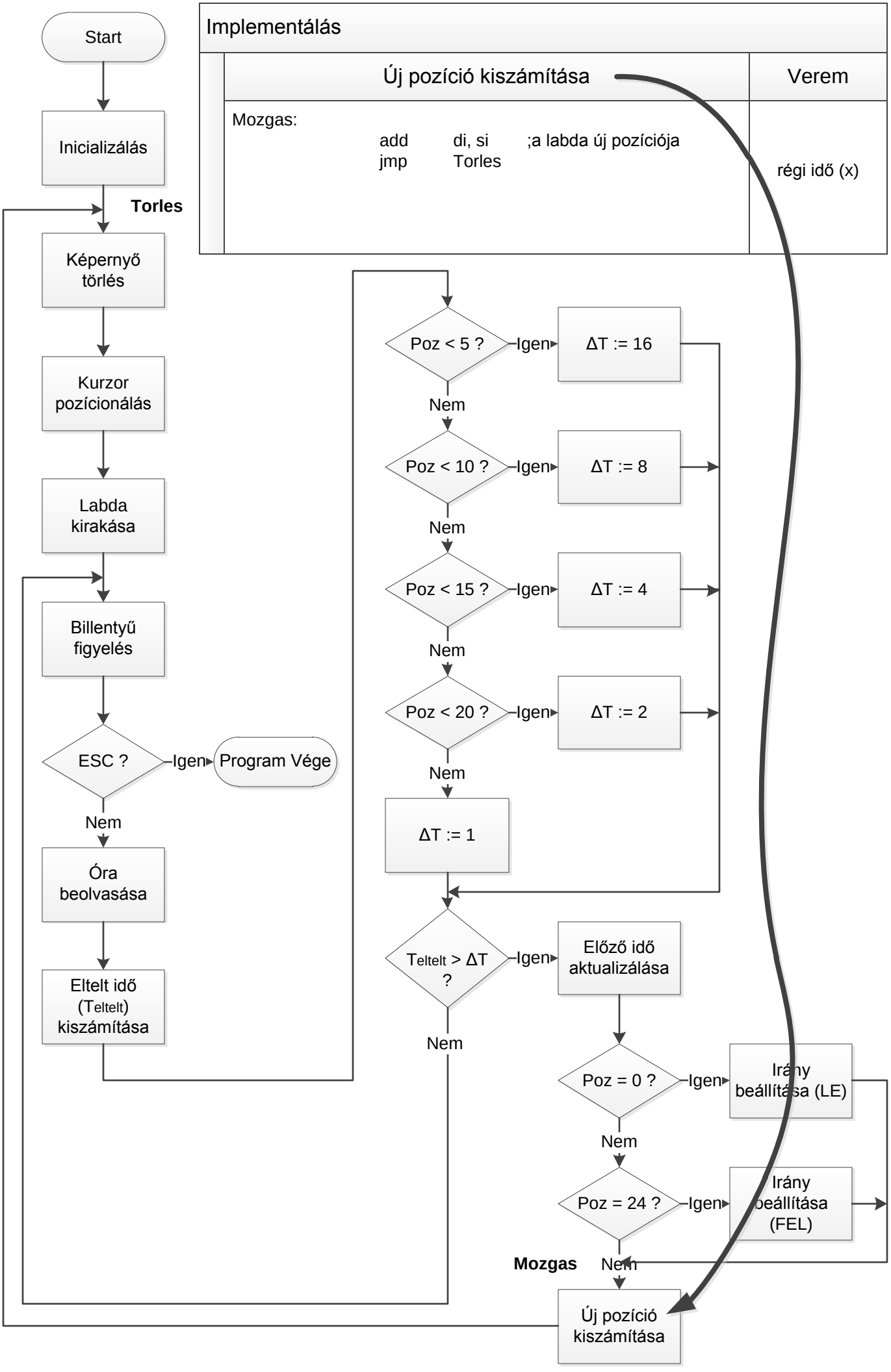

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
ΔT := 1
Verem
Ido4:
mov al, 1 ;Egyébként (21-24)
aktuális idő
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

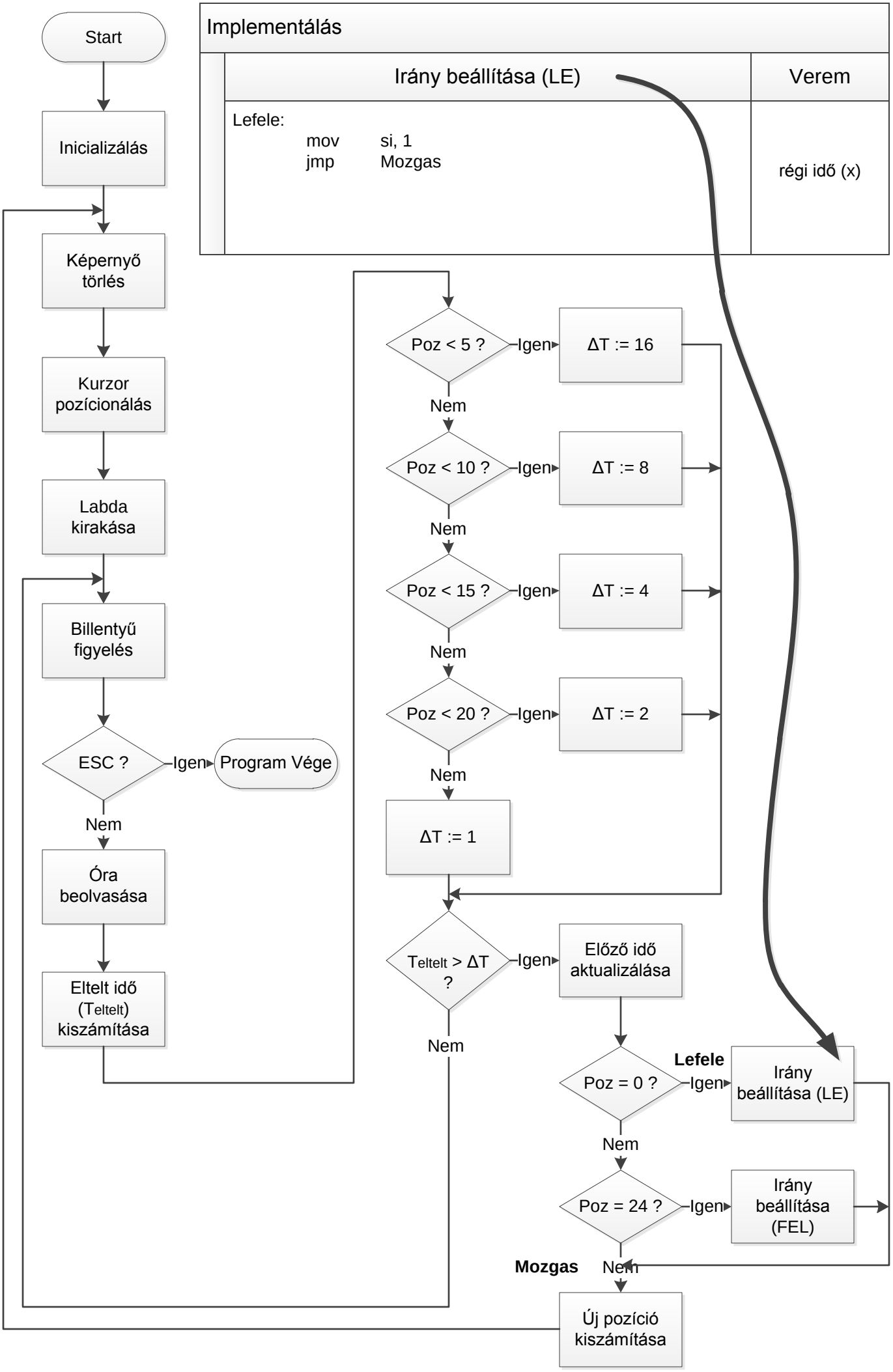

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Kesleltet
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Teltelt > ΔT ?
Verem
Beallit:
xor ah, ah
cmp dx, ax
pop ax ;előző idő aktualizálásához kell
jc Kesleltet
régi idő (0)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Beállít
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

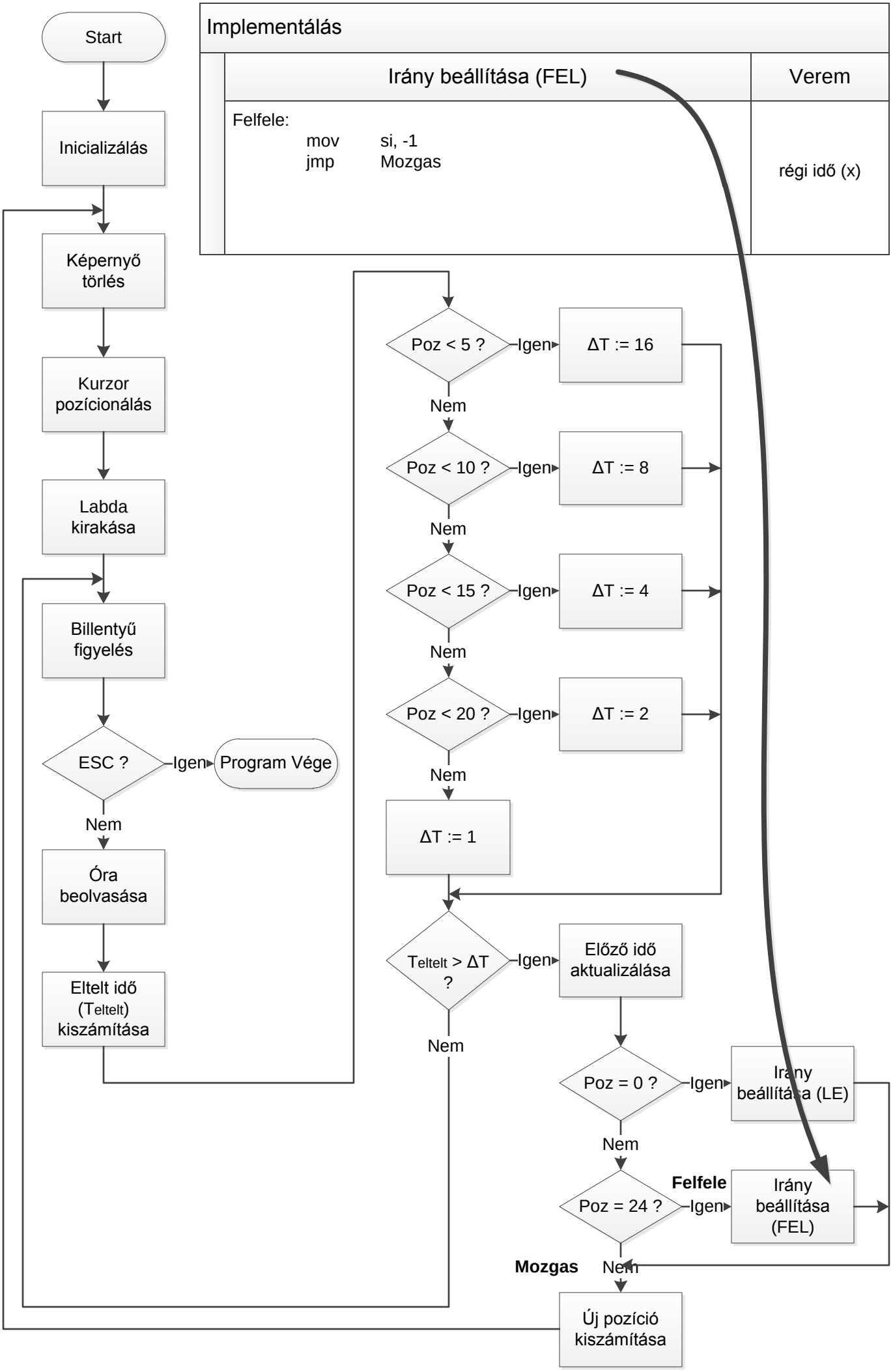

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Előző idő aktualizálása
Verem
pop cx ;régi idő kivétele a veremből
push ax ;aktuális idő elmentése
régi idő (x)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása

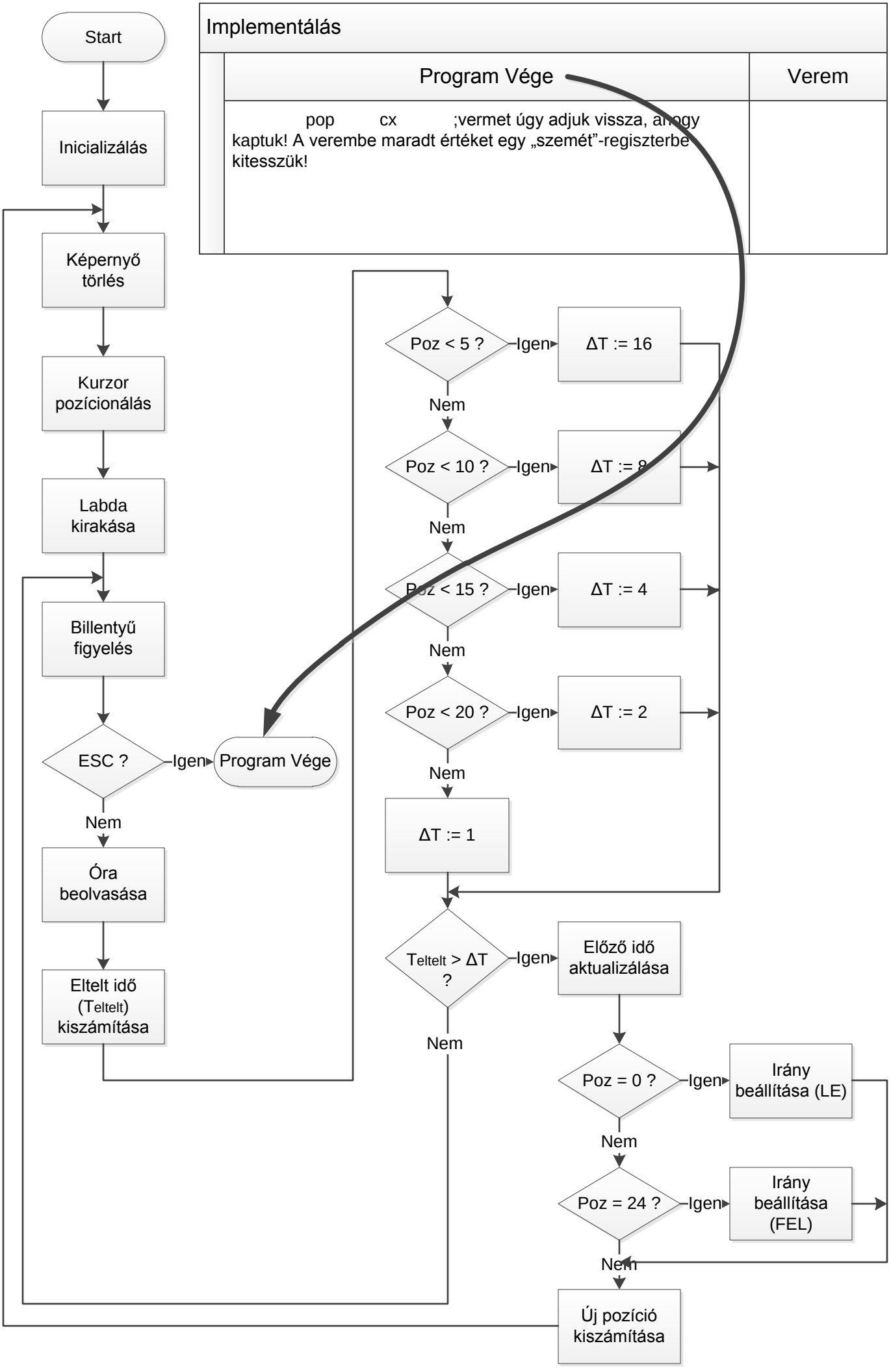

Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Poz = 0 ?
Verem
cmp di, 0
jz Lefele
régi idő (x)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Lefele
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása


Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Poz = 24 ?
Verem
cmp di, 24
jz Felfele
régi idő (x)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Felfele
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása


Start
Inicializálás
Torles
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Új pozíció kiszámítása
Verem
Mozgas:
add di, si ;a labda új pozíciója
jmp Torles
régi idő (x)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Mozgas
Nem
Új pozíció kiszámítása


Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Irány beállítása (LE)
Verem
Lefele:
mov si, 1
jmp Mozgas
régi idő (x)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Lefele
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Mozgas
Nem
Új pozíció kiszámítása


Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Irány beállítása (FEL)
Verem
Felfele:
mov si, -1
jmp Mozgas
régi idő (x)
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Felfele
Irány beállítása (FEL)
Mozgas
Nem
Új pozíció kiszámítása


Start
Inicializálás
Képernyő törlés
Kurzor pozícionálás
Labda kirakása
Billentyű figyelés
ESC ?
Igen
Program Vége
Nem
Óra beolvasása
Eltelt idő (Teltelt) kiszámítása
Implementálás
Program Vége
Verem
pop cx ;vermet úgy adjuk vissza, ahogy kaptuk! A verembe maradt értéket egy „szemét”-regiszterbe kitesszük!
Poz < 5 ?
Igen
ΔT := 16
Nem
Poz < 10 ?
Igen
ΔT := 8
Nem
Poz < 15 ?
Igen
ΔT := 4
Nem
Poz < 20 ?
Igen
ΔT := 2
Nem
ΔT := 1
Teltelt > ΔT ?
Igen
Előző idő aktualizálása
Nem
Poz = 0 ?
Igen
Irány beállítása (LE)
Nem
Poz = 24 ?
Igen
Irány beállítása (FEL)
Nem
Új pozíció kiszámítása