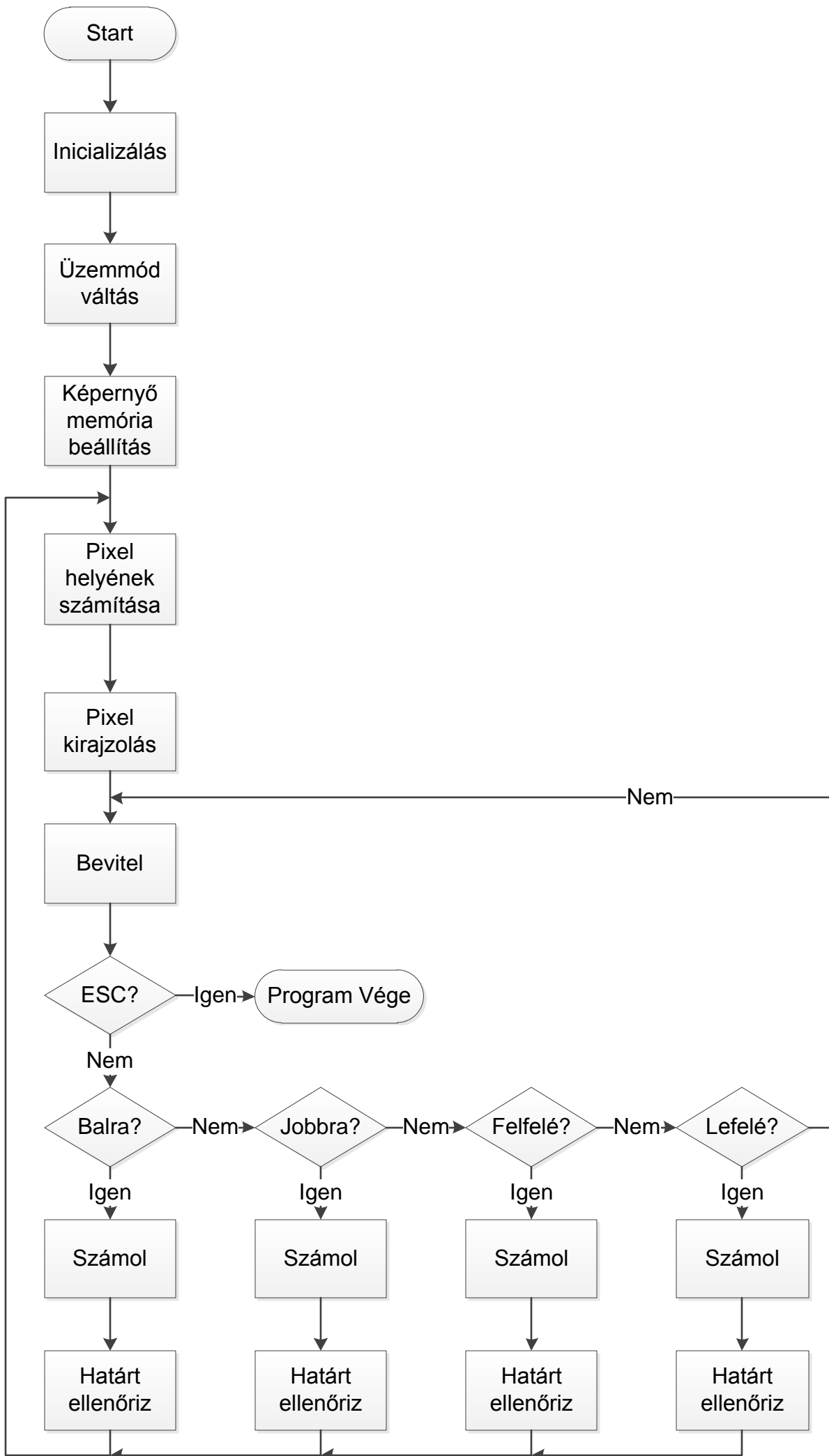


Feladat5: rajzolo.asm

Feladat meghatározása	Implementálás	Implementálás
<p>A feladat célja bemutatni egy egyszerű grafikai program segítségével a közvetlen memóriacímzést (grafikus VGA 320*200).</p> <p>A program a kurzor mozgató gombok segítségével folytonos vonalat rajzol miközben figyeli a rajzolásra kijelölt terület határait.</p> <div style="text-align: center; margin-top: 20px;"> <p>8 bites megoldás</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <pre> pop    dx push   dx xor    dh, dh add    ax, dx Pixel: mov    di, ax                     </pre> </div> <p>16 bites megoldás</p> </div>	<pre> Code    Segment         assume CS:Code,         DS:Data, SS:Stack  Start:         mov     ax, Code         mov     DS, AX          mov     dl, 100         mov     dh, 100         push    dx          mov     ax, 13h         int     10h          mov     ax, 0a000h         mov     es, ax  Rajz:         pop     dx         xor     ah, ah         mov     push  al, dh         push   dx         mov     bx, 320         mul    bx         pop     dx         add    al, dl         jnc    Pixel         inc   ah  Pixel:         push   dx         mov    di, ax         mov    al, 128         mov    es:[di], al  Var:         xor    ah, ah         int   16h          cmp    al, 27         jz    Program_Vege          cmp    ah, 75         jz    Balra          cmp    ah, 77         jz    Jobbra          cmp    ah, 72         jz    Felfele          cmp    ah, 80         jz    Lefele          jmp   Var                     </pre>	<pre> Balra:         pop     dx         dec    dl         cmp    dl, 1         ;      jnc    Tarol1         jnc    Tarol         inc    dl         jmp    Tarol  ;Tarol1:         ;      push   dx         ;      jmp    Rajz  Jobbra:         pop     dx         inc    dl         cmp    dl, 250         ;      jc     Tarol2         jc     Tarol         dec    dl         jmp    Tarol  ;Tarol2:         ;      push   dx         ;      jmp    Rajz  Felfele:         pop     dx         dec    dh         cmp    dh, 1         ;      jnc    Tarol3         jnc    Tarol         inc    dh         jmp    Tarol  ;Tarol3:         ;      push   dx         ;      jmp    Rajz  Lefele:         pop     dx         inc    dh         cmp    dh, 200         ;      jc     Tarol4         jc     Tarol         dec    dh         jmp    Tarol  ;Tarol4:         ;      push   dx         ;      jmp    Rajz  Tarol:         push   dx         jmp   Rajz  Program_Vege:         mov    ax, 03h         int   10h          pop    dx          mov    ax, 4c00h         int   21h  Code    Ends Data    Segment Data    Ends Stack   Segment Stack   Ends End     Start                     </pre>



Start

Inicializálás

Üzem mód váltás

Képernyő memória beállítás

Pixel helyének számítása

Pixel kirajzolás

Bevitel

ESC?

Program Vége

Balra?

Jobbra?

Felfelé?

Lefelé?

Számol

Számol

Számol

Számol

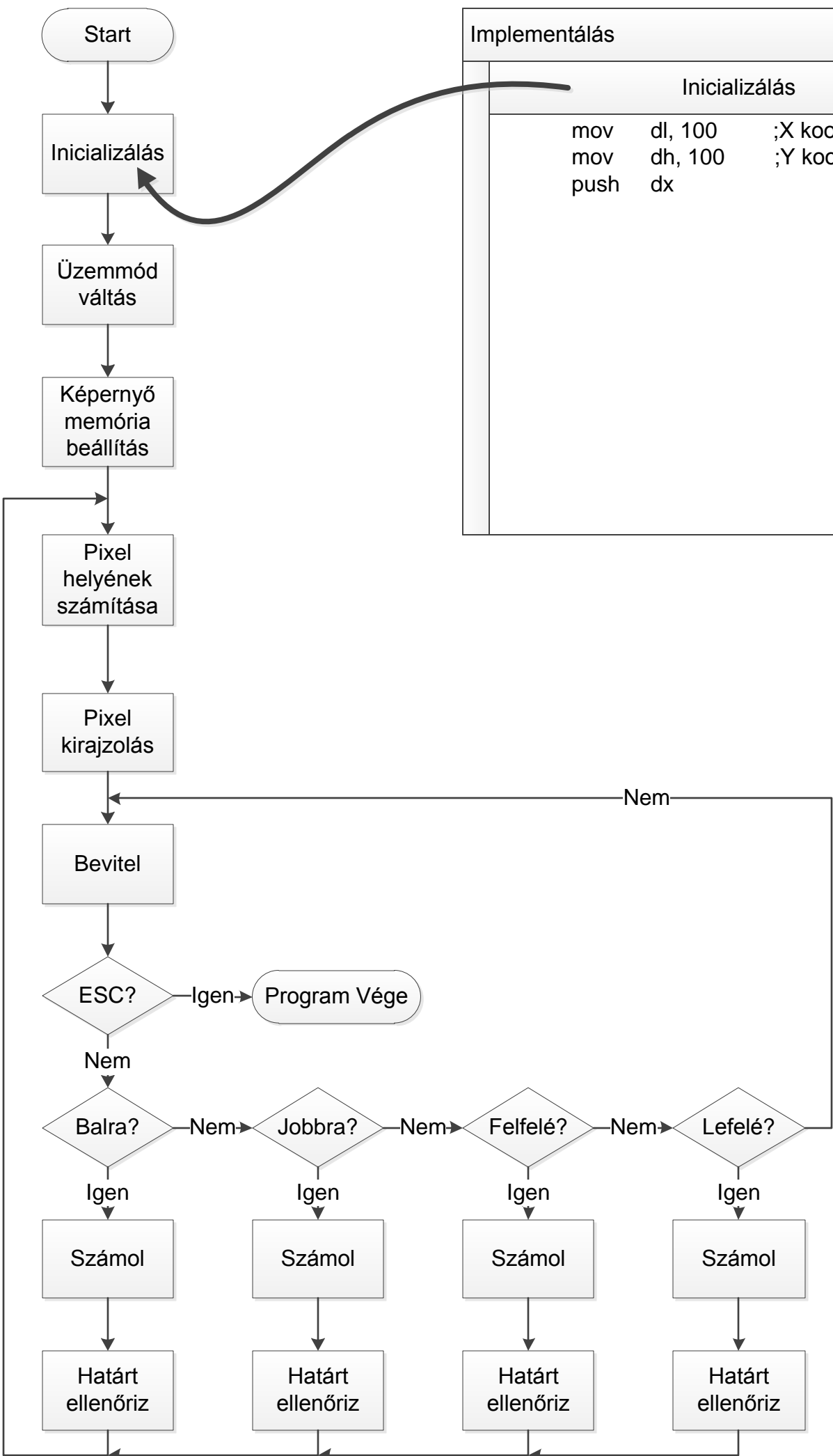
Határt ellenőriz

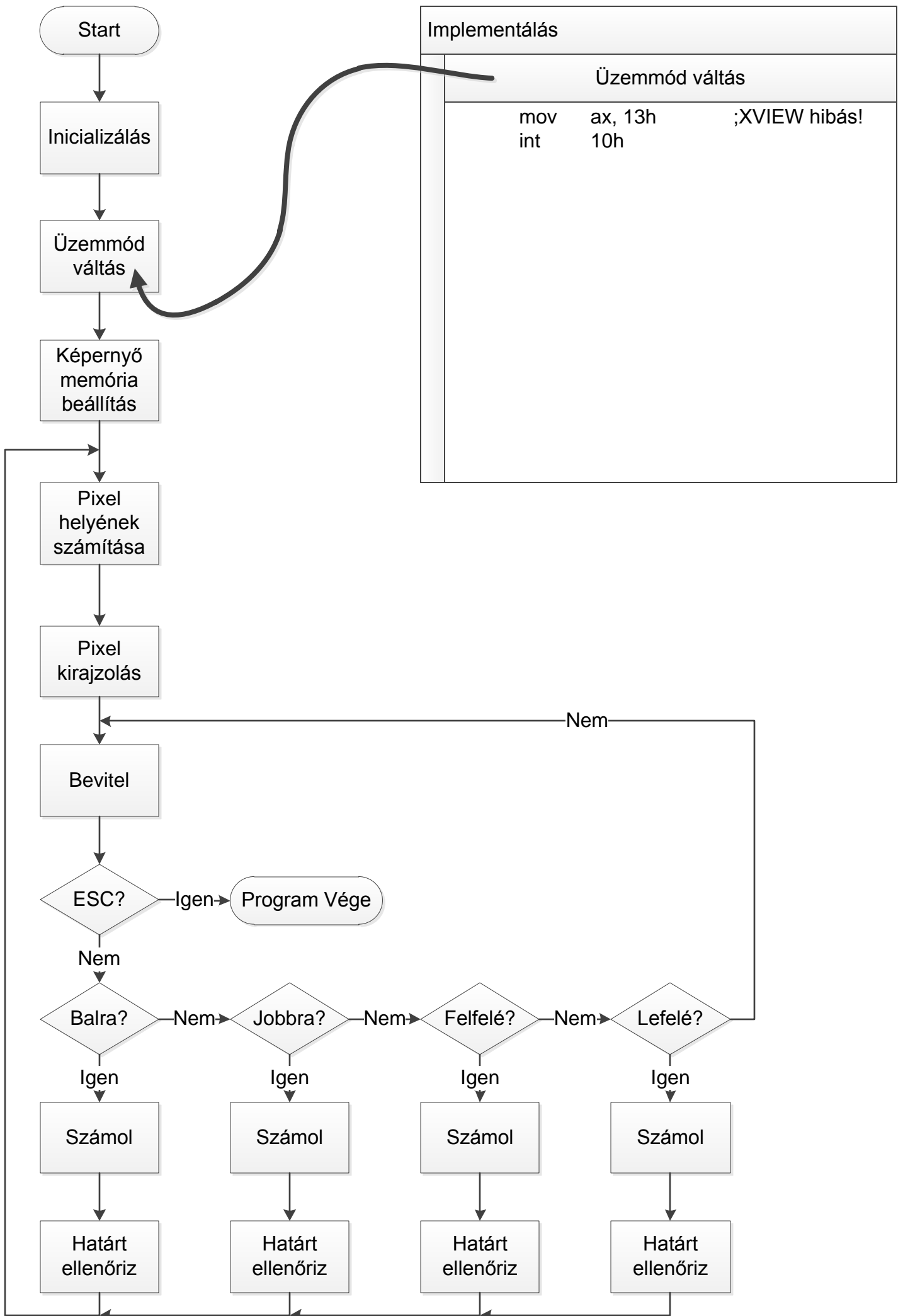
Határt ellenőriz

Határt ellenőriz

Határt ellenőriz

Implementálás		
Inicializálás		
mov	dl, 100	;X koordináta
mov	dh, 100	;Y koordináta
push	dx	





Start

Inicializálás

Üzem mód váltás

Képernyő memória beállítás

Pixel helyének számítása

Pixel kirajzolás

Bevitel

ESC?

Igen → Program Vége

Nem

Balra?

Nem → Jobbra?

Igen

Számol

Határt ellenőriz

Jobbra?

Nem → Felfelé?

Igen

Számol

Határt ellenőriz

Felfelé?

Nem → Lefelé?

Igen

Számol

Határt ellenőriz

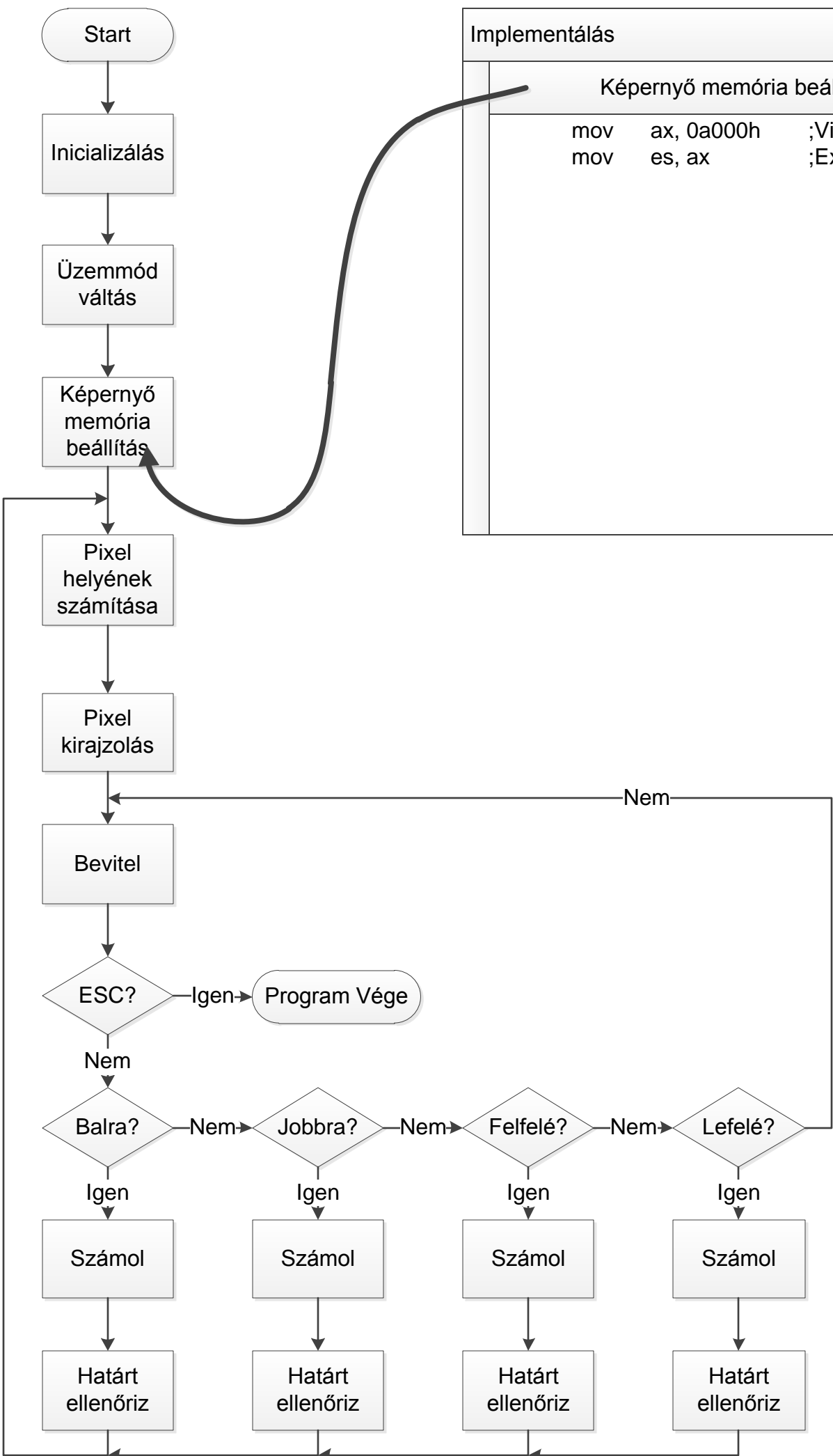
Lefelé?

Igen

Számol

Határt ellenőriz

Implementálás		
Képernyő memória beállítás		
mov	ax, 0a000h	;Videó Kezdőcím
mov	es, ax	;Extra Segmens



Start

Inicializálás

Üzem mód váltás

Képernyő memória beállítás

Rajz

Pixel helyének számítása

Pixel

Pixel kirajzolás

Bevitel

ESC?

Igen

Program Vége

Nem

Balra?

Nem

Jobbra?

Nem

Felfelé?

Nem

Lefelé?

Igen

Számol

Határt ellenőriz

Igen

Számol

Határt ellenőriz

Igen

Számol

Határt ellenőriz

Igen

Számol

Határt ellenőriz

Nem

### Implementálás

#### Pixel helyének számítása

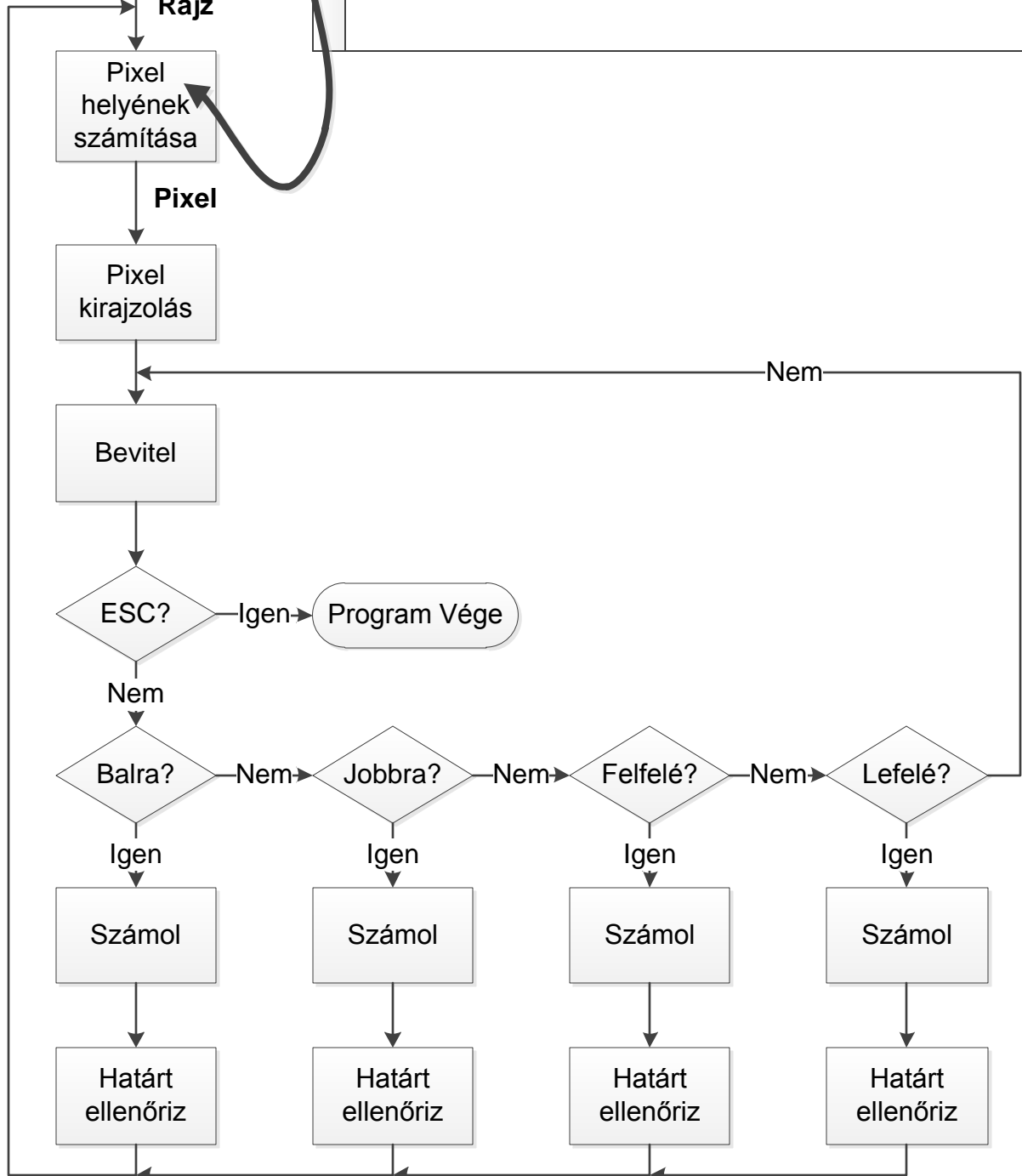
; Pixel = Y \* 320 + X

Rajz:

```

pop    dx           ;dx-ben Y(dh), X(dl) koordináta
xor    ah, ah
mov    al, dh       ;ax-ben Y koordináta
push  dx           ;dx mentése, mul utasítás felülírja
mov    bx, 320
mul   bx           ;Y koordináta * 320
pop    dx           ;dx-ben Y(dh), X(dl) koordináta
add   al, dl       ;X koordináta hozzáadása
jnc   Pixel
inc   ah           ;van átvitel

```



Start

Inicializálás

Üzem mód váltás

Képernyő memória beállítás

Pixel helyének számítása

Pixel

Pixel kirajzolás

Bevitel

ESC?

Igen

Program Vége

Nem

Balra?

Nem

Jobbra?

Nem

Felfelé?

Nem

Lefelé?

Igen

Igen

Igen

Igen

Számol

Számol

Számol

Számol

Határt ellenőriz

Határt ellenőriz

Határt ellenőriz

Határt ellenőriz

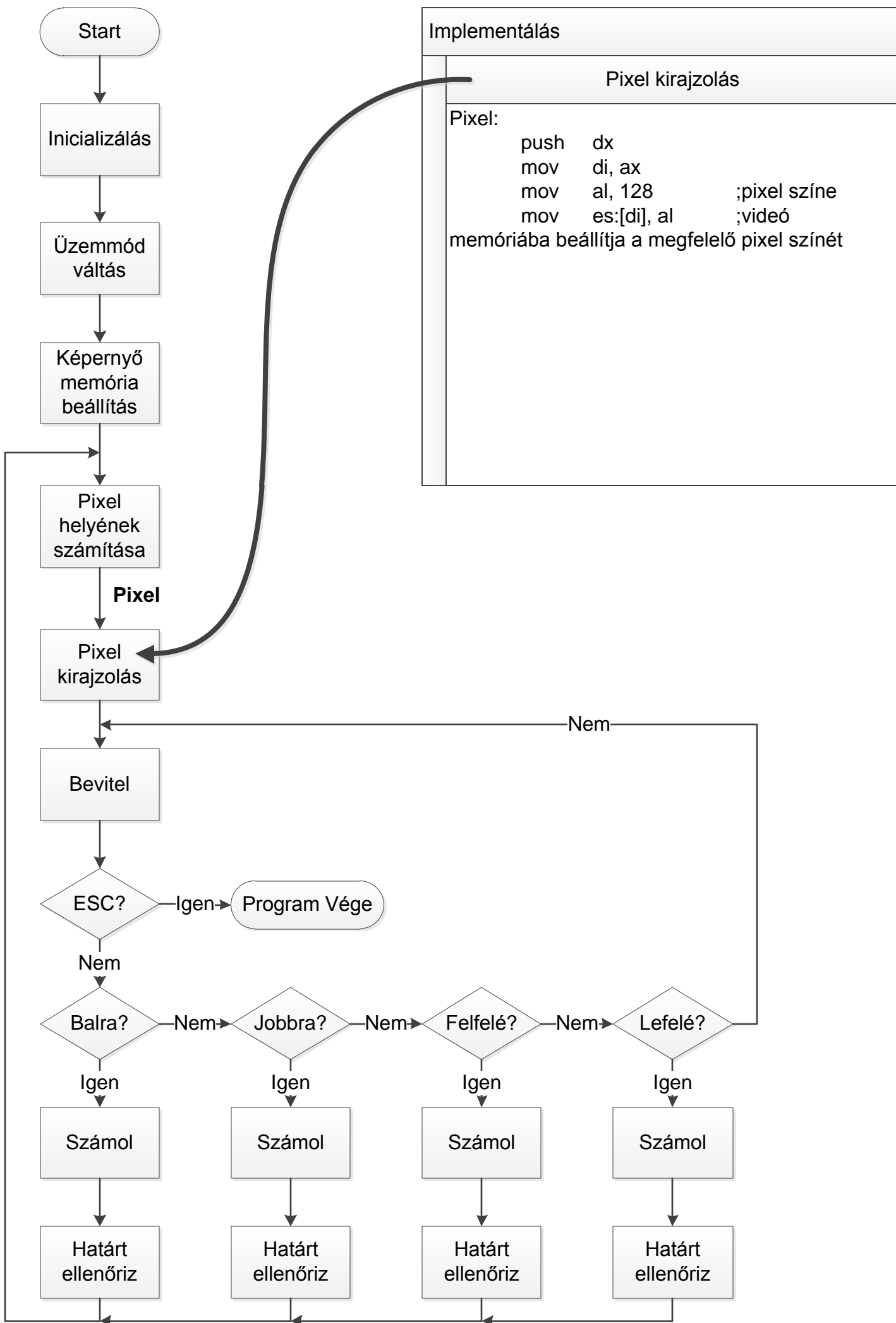
### Implementálás

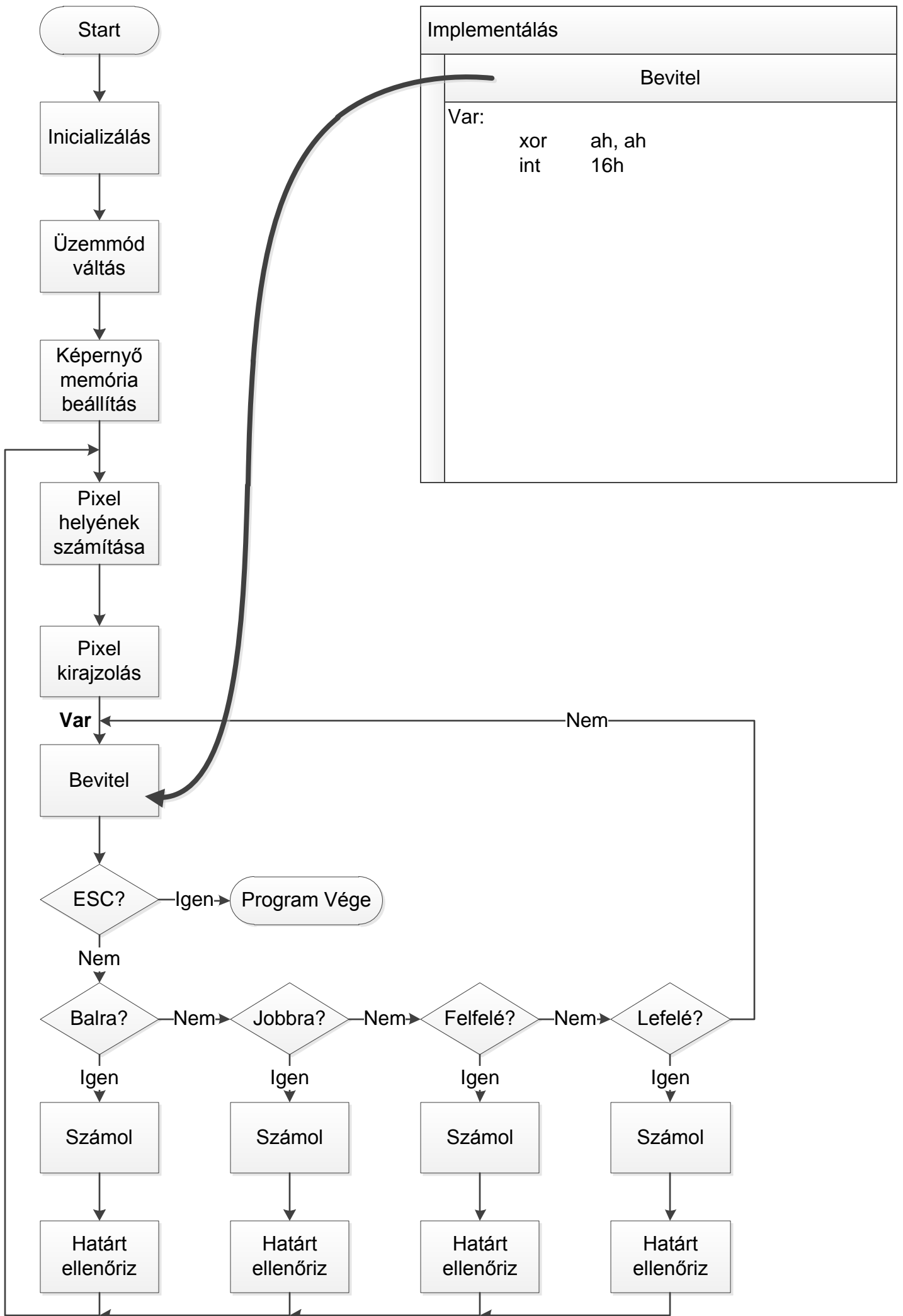
#### Pixel kirajzolás

Pixel:

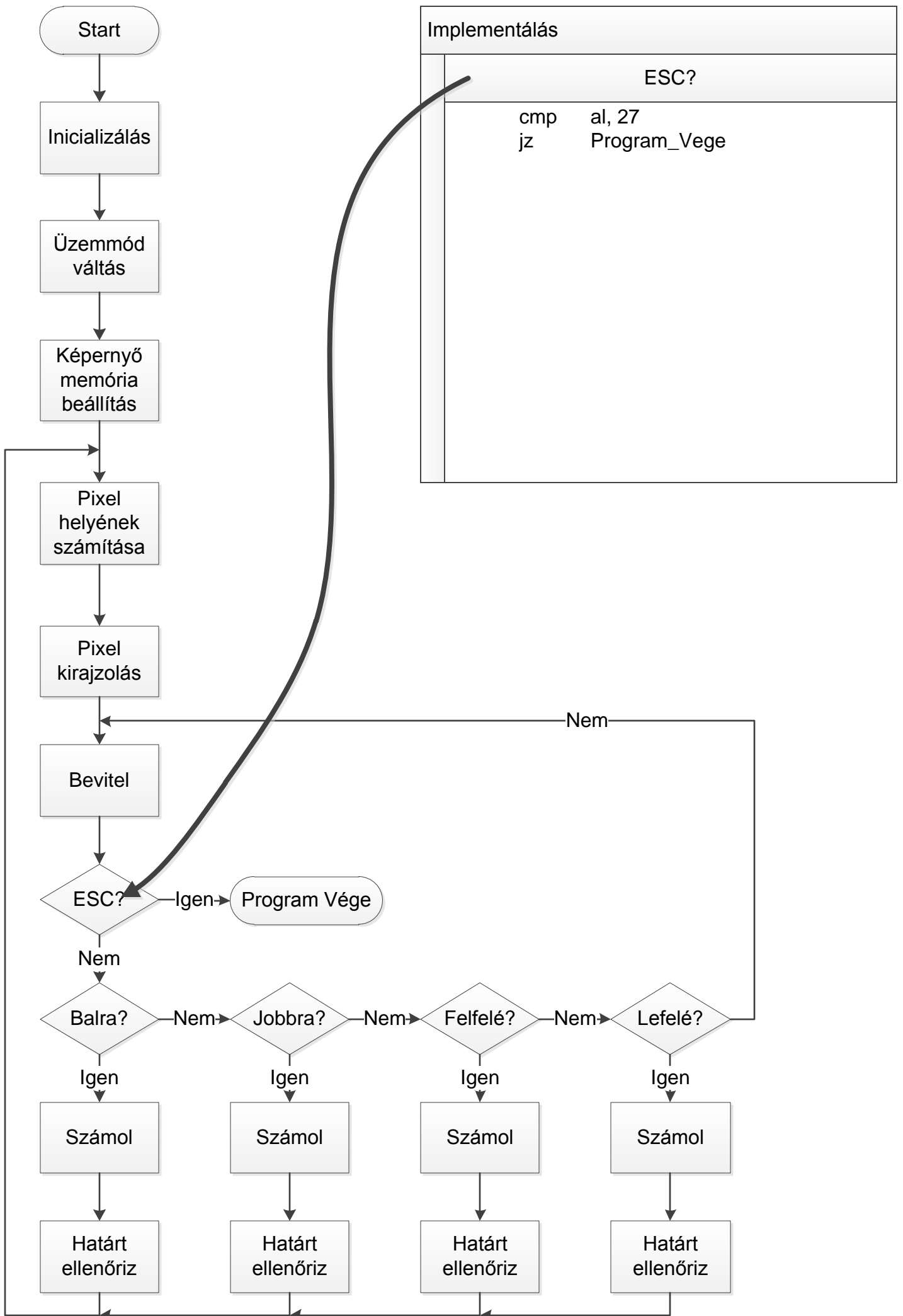
```
push dx
mov di, ax
mov al, 128 ;pixel színe
mov es:[di], al ;videó
```

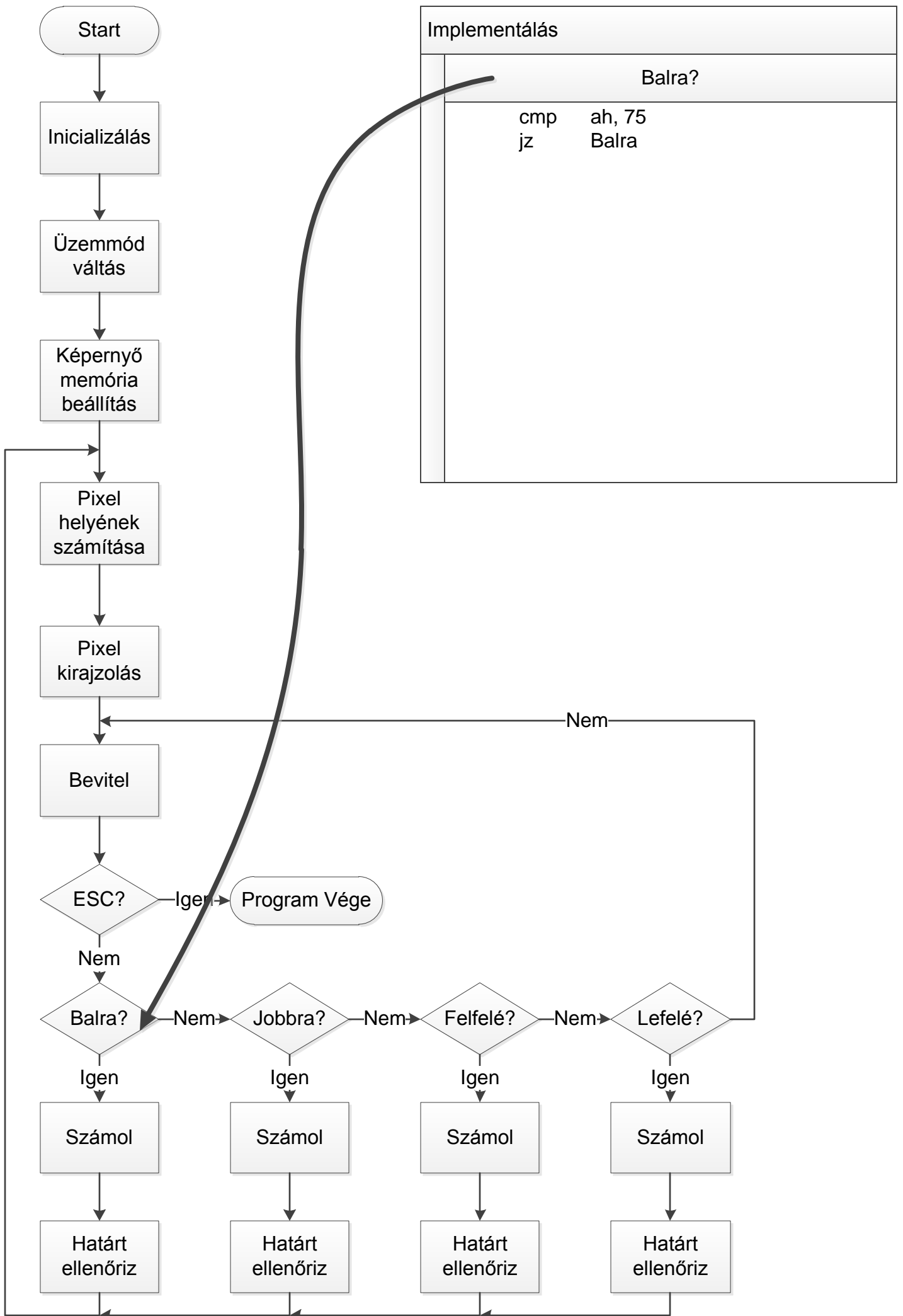
memóriába beállítja a megfelelő pixel színét

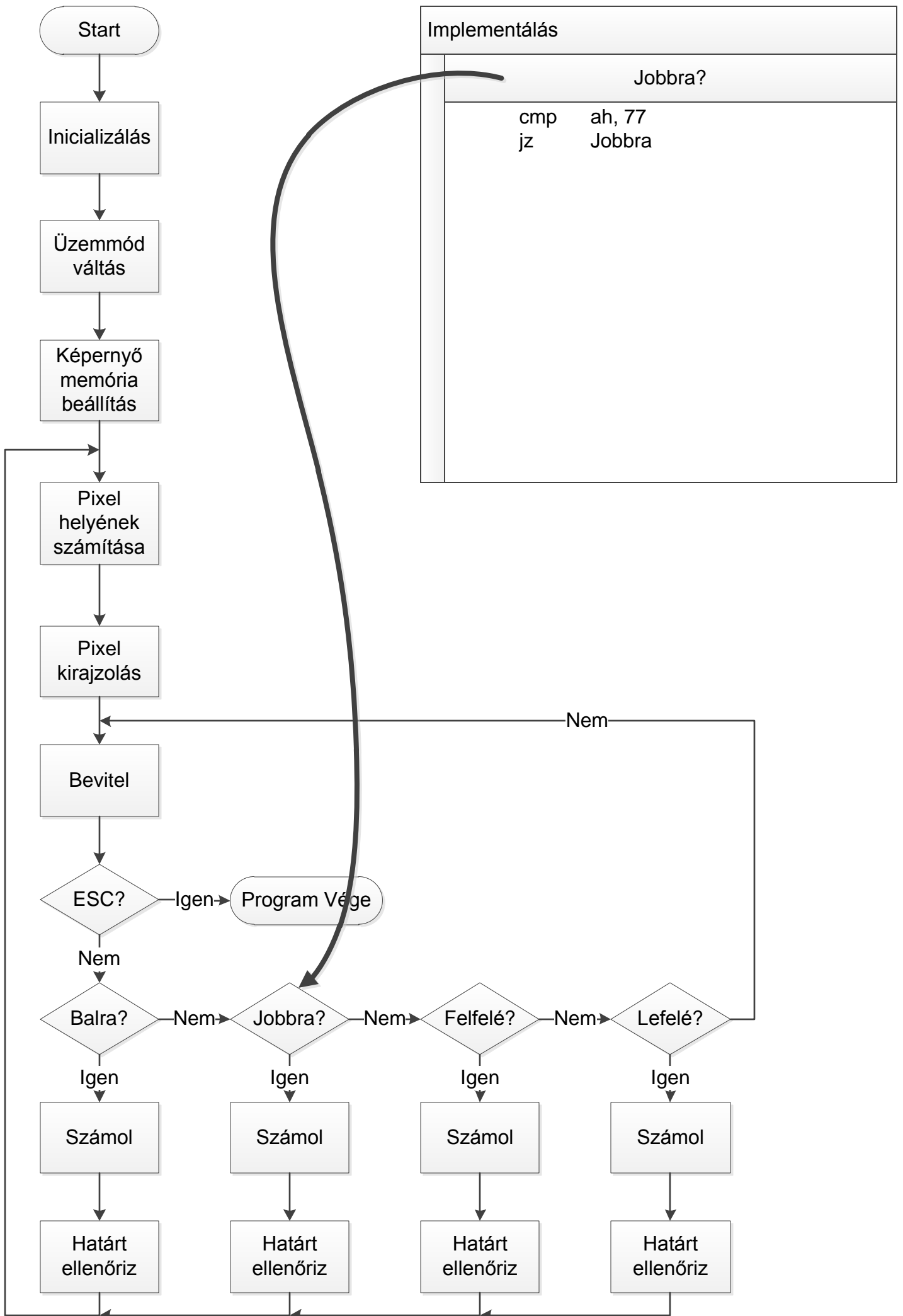


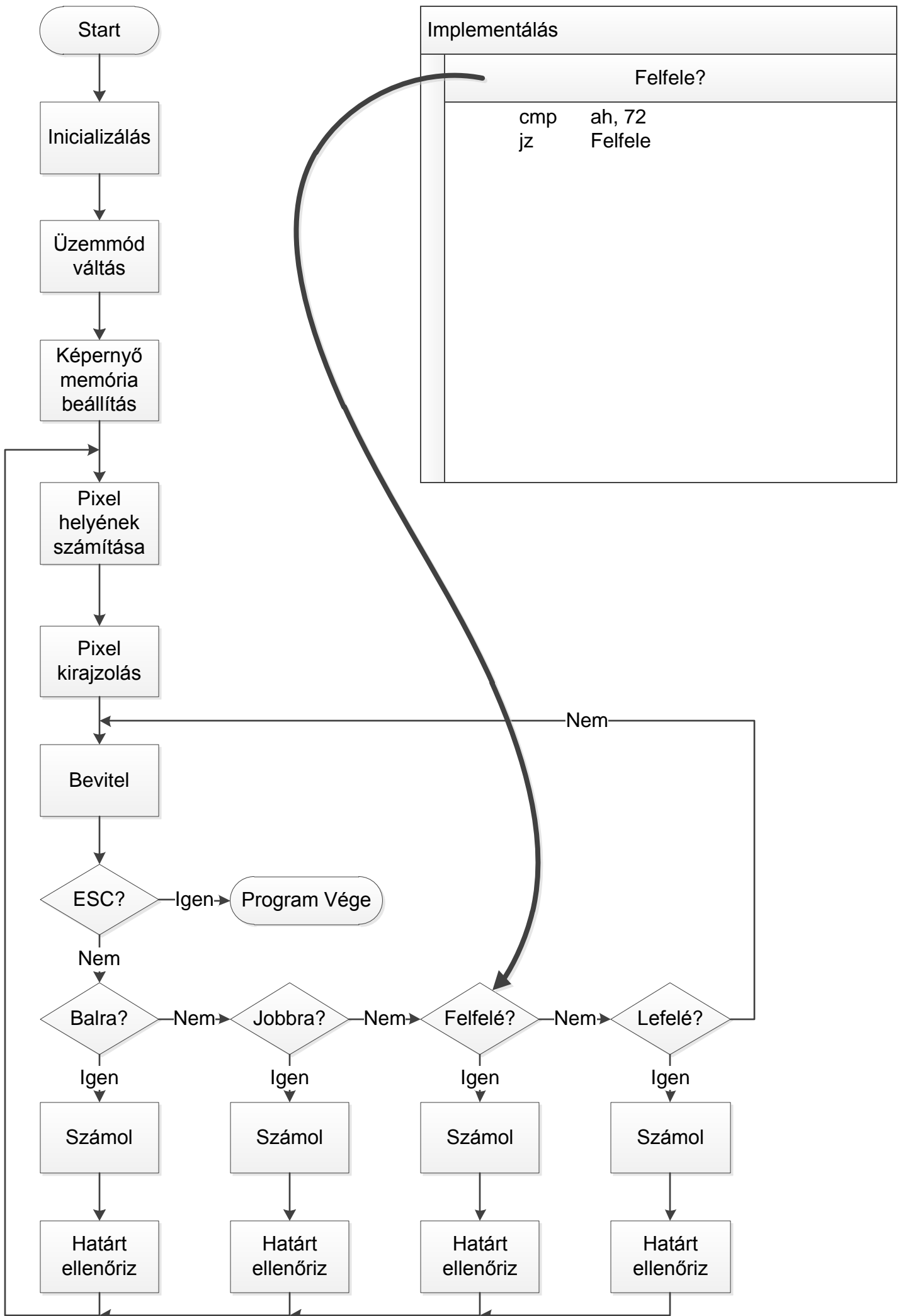




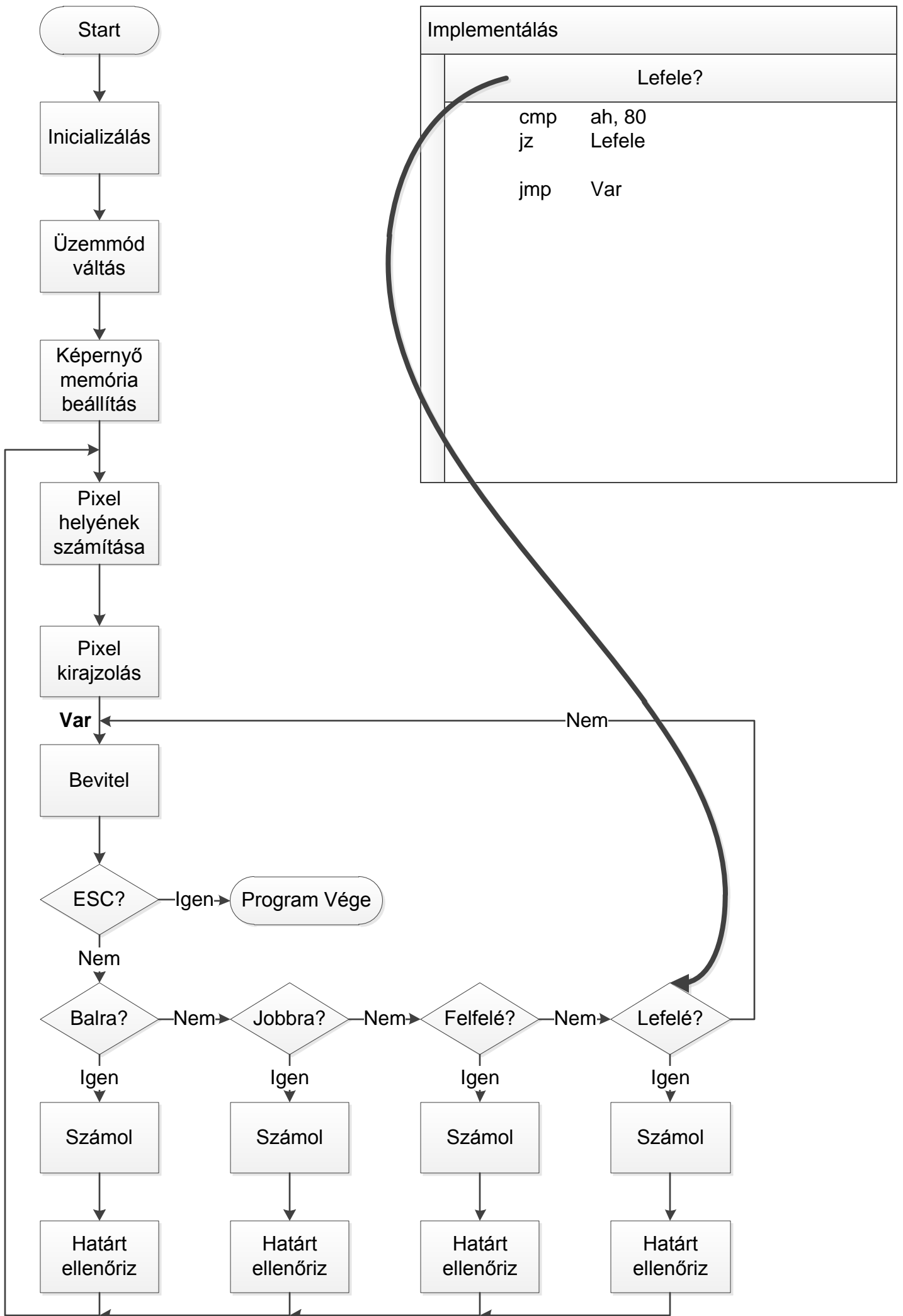


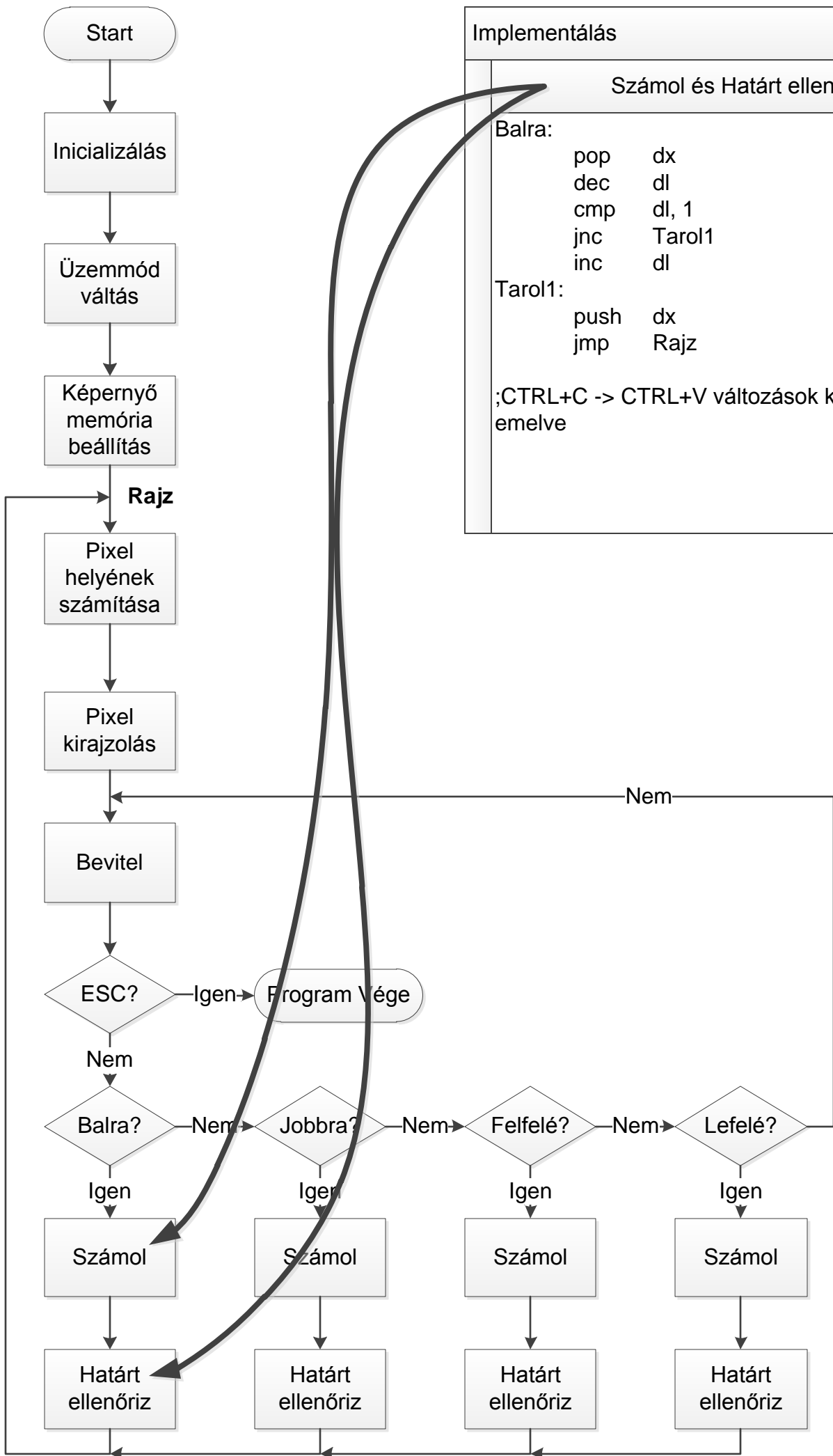




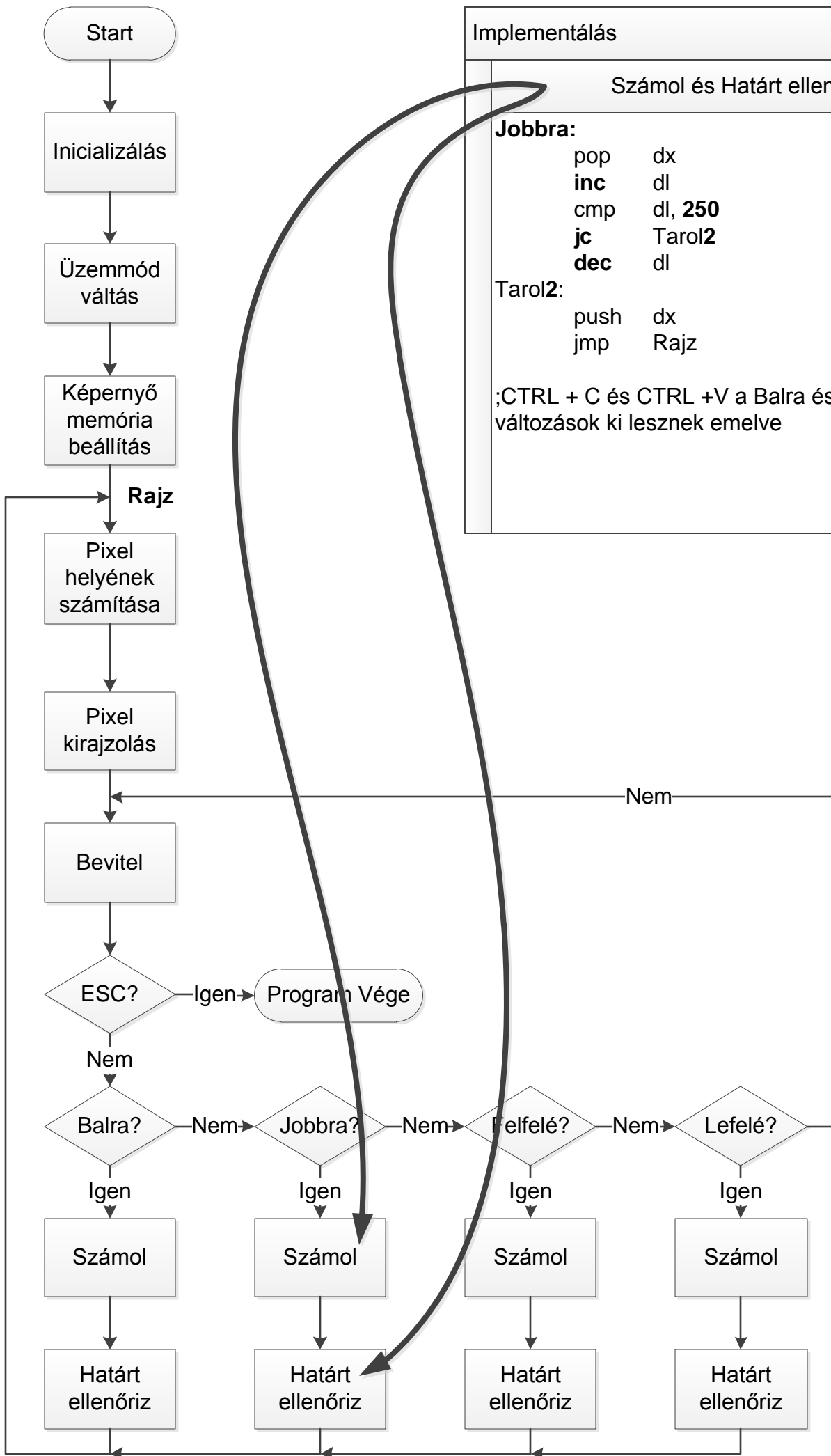


Implementálás	
Felfele?	
cmp	ah, 72
jz	Felfele

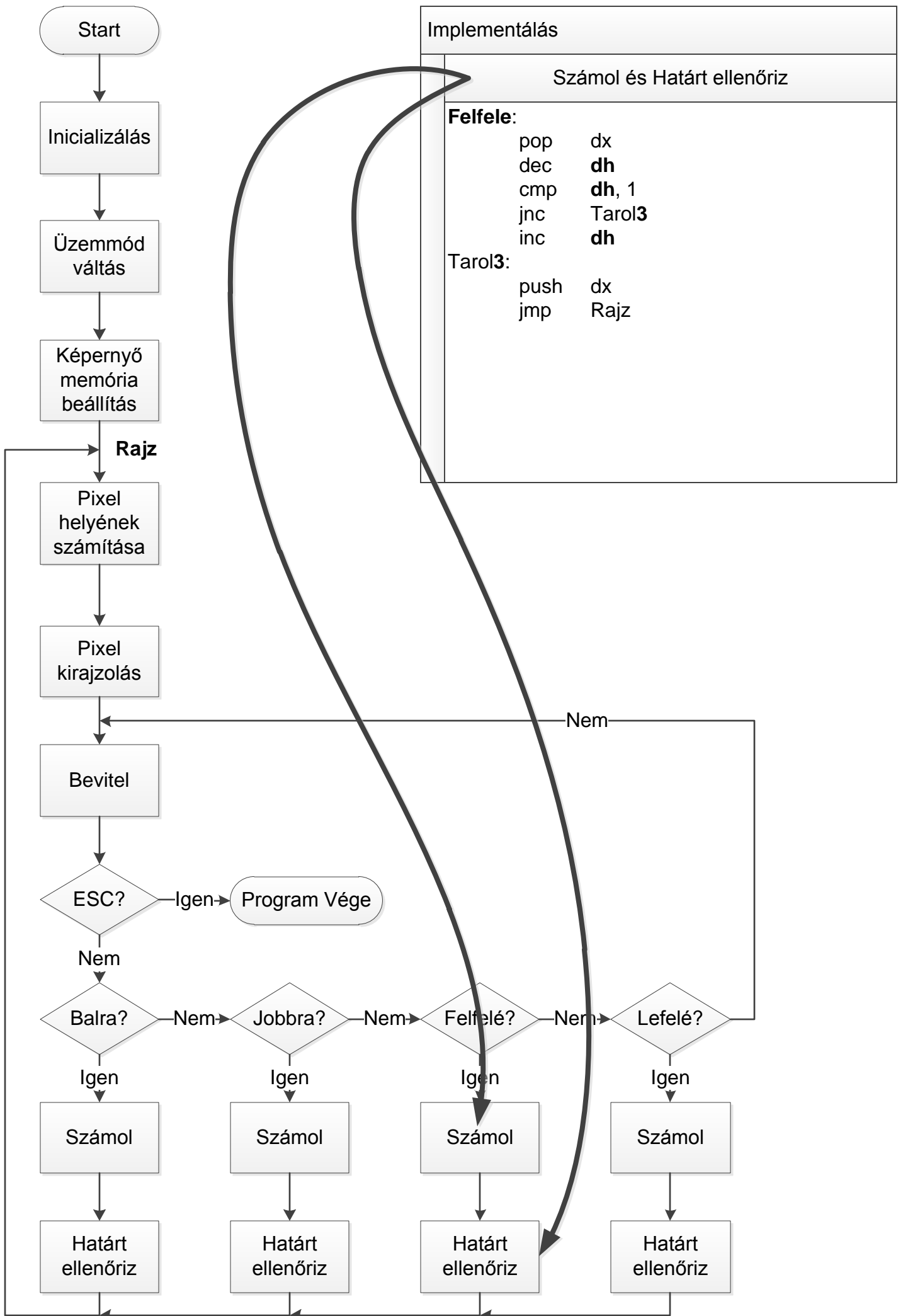




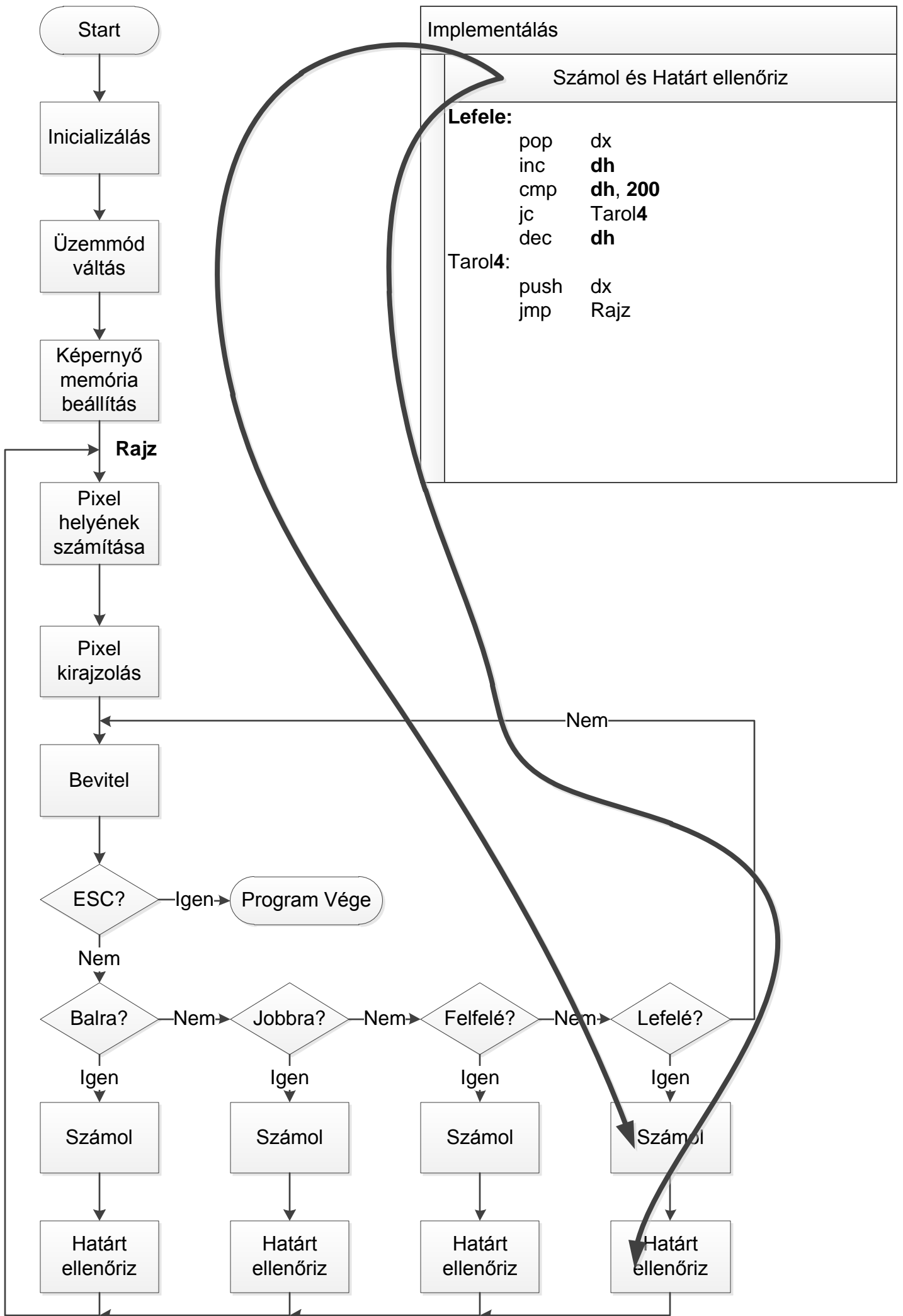
Implementálás	
Számol és Határt ellenőriz	
Balra:	<pre> pop    dx dec    dl cmp    dl, 1 jnc    Tarol1 inc    dl </pre>
Tarol1:	<pre> push   dx jmp    Rajz </pre>
;CTRL+C -> CTRL+V változások ki lesznek emelve	

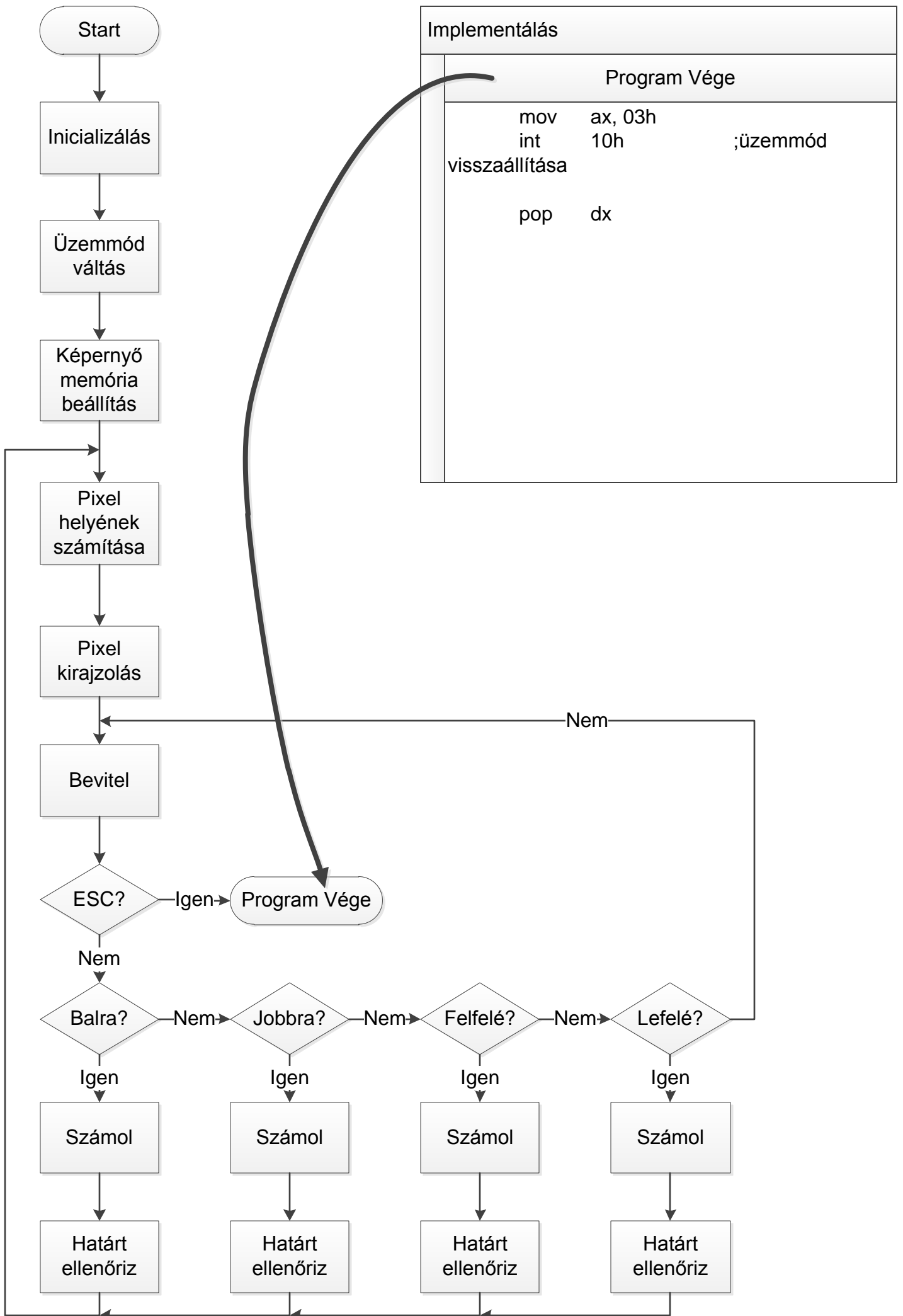


Implementálás	
Számol és Határt ellenőriz	
<b>Jobbra:</b>	
pop	dx
<b>inc</b>	di
cmp	di, 250
<b>jc</b>	Tarol2
<b>dec</b>	di
<b>Tarol2:</b>	
push	dx
jmp	Rajz
;CTRL + C és CTRL +V a Balra és Jobbra blokk, változások ki lesznek emelve	









Implementálás			
Program Vége			
mov	ax,	03h	
int	10h		;üzemmód
visszaállítása			
pop	dx		