

DIGITÁLIS TECHNIKA

8

Dr. Oniga István

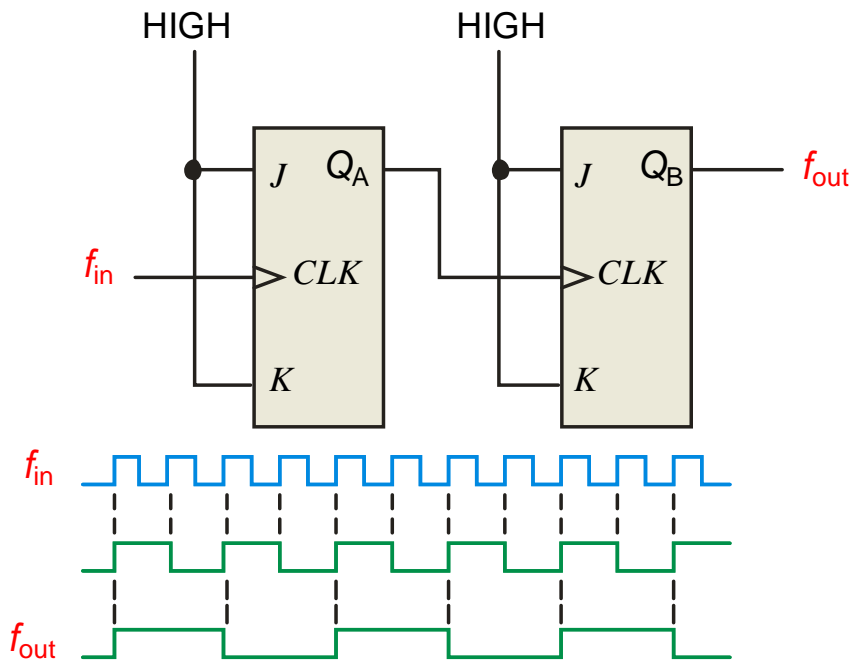
Szekvenciális (sorrendi) hálózatok

- Szekvenciális hálózatok fogalma
- Tárolók
 - RS tárolók
 - JK tárolók
 - T és D típusú tárolók
- **Számlálók**
 - Szinkron számlálók
 - Aszinkron számlálók
- Regiszterek
- <http://www.play-hookey.com/digital/>
- <http://www.asic-world.com/digital/seq.html>

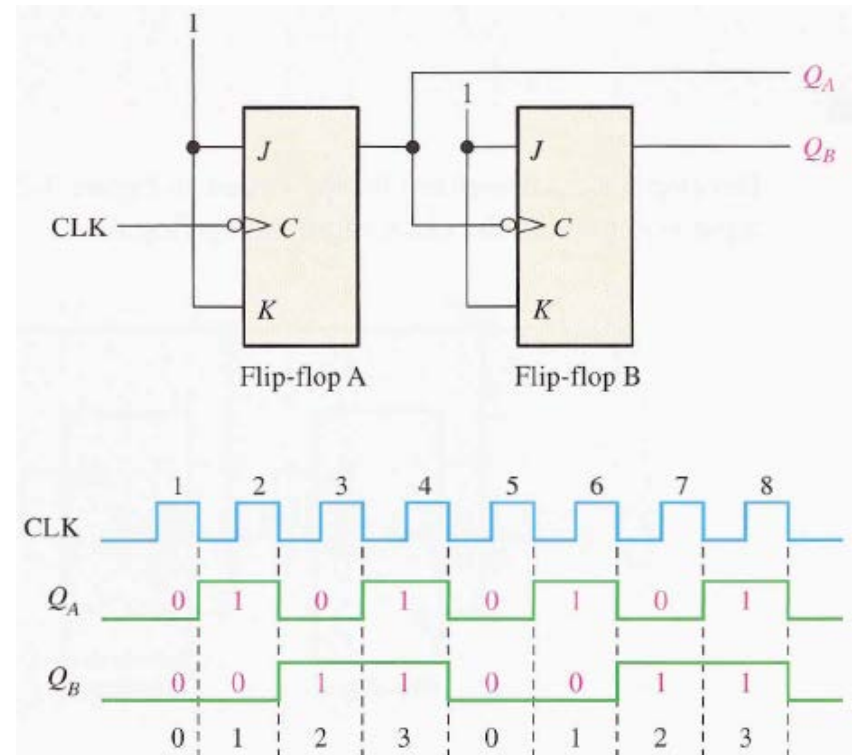
A tárolók alkalmazásai

- Minden felfutó élre a kimenet az előző állapot negáltjára vált
- Egy flip-flop kettővel osztja a bemenőjelet, két flip-flops négyel

Frekvenciaosztás



Számlálók



Számlálók

A számlálók - szekvenciális áramkörök, amelyek összeszámlálják a bemenetükre érkező impulzusokat, és az eredményt a Q kimeneteken jelenítik meg.

◆ Számlálás iránya szerint:

- felfelé (előre) számlálók
- lefelé számlálók
- fel/le(reverzibilis) számlálók

◆ Számlálás kódja szerint:

- bináris
- BCD

◆ Vezérlési mód szempontjából:

- aszinkron
- szinkron

◆ Egyéb szolgáltatások:

- szinkron/aszinkron törlés
- szinkron aszinkron kezdőérték beállítás (programozhatóság)

Aszinkron számlálók

- Minden egyes T tároló felezi a bemenetére kapcsolt jel frekvenciáját, az egyik periódusában felvesz egy értéket, a másik periódusban ennek negáltját.
- T tárolókat sorba kötünk $\Rightarrow n \times TFF$, 2^n osztást végez.
- A kapacitás azt a legnagyobb számot jelenti, amellyel egy számolási ciklus befejeződik.
- A számláló $m = k + 1$ különböző értéket különböztet meg.
- Az m - et a számláló modulusának nevezzük.
- Pl. 4 TFF – 0-15 számol $\Rightarrow m=16$

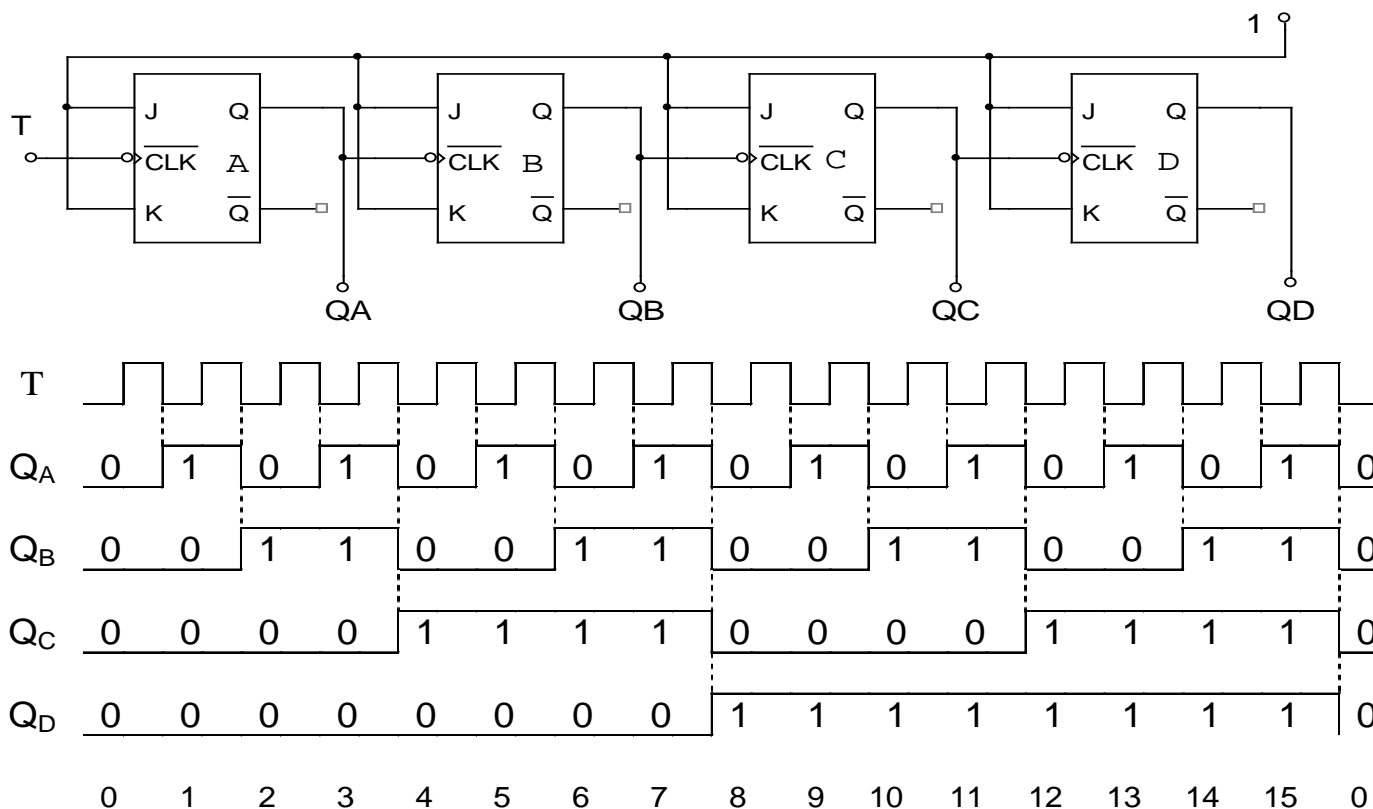
Nr. Imp.tact	Q _D	Q _C	Q _B	Q _A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16 (0)	0	0	0	0
17 (1)	0	0	0	1

Az *aszinkron működés* = a tárolók nem szinkronban „billenek”, a *számlálódó* jel elindítja a állapotváltozási sorozatot. A továbbiakban az egyes *FF- ok billentik egymást*.

\Rightarrow rövid ideig hibás kombináció jelenik meg a kimeneteken

Aszinkron felfelé számlálók

- T tárolókat úgy kötünk sorba, hogy a Q kimeneteket kötjük a következő T-k órajel bemenetére.
- Előreszámlálás akkor valósul meg, ha lefutó élre vezérelt tárolókat alkalmazunk



$$N_x = Q_D * 2^3 + Q_C * 2^2 + Q_B * 2^1 + Q_A * 2^0$$

Aszinkron felfelé számlálók jellemzői

- Az átmeneti időszakban nem *kívánt kombinációk is előfordulnak a kimeneteken*
- Az aszinkron számlálók nagyon egyszerű felépítése mellett hátránya a kisebb határfrekvencia.
- Ha egy FF terjedési ideje tipikusan $T_p = 10$ ns akkor egy négybites aszinkron számláló teljes terjedési ideje:

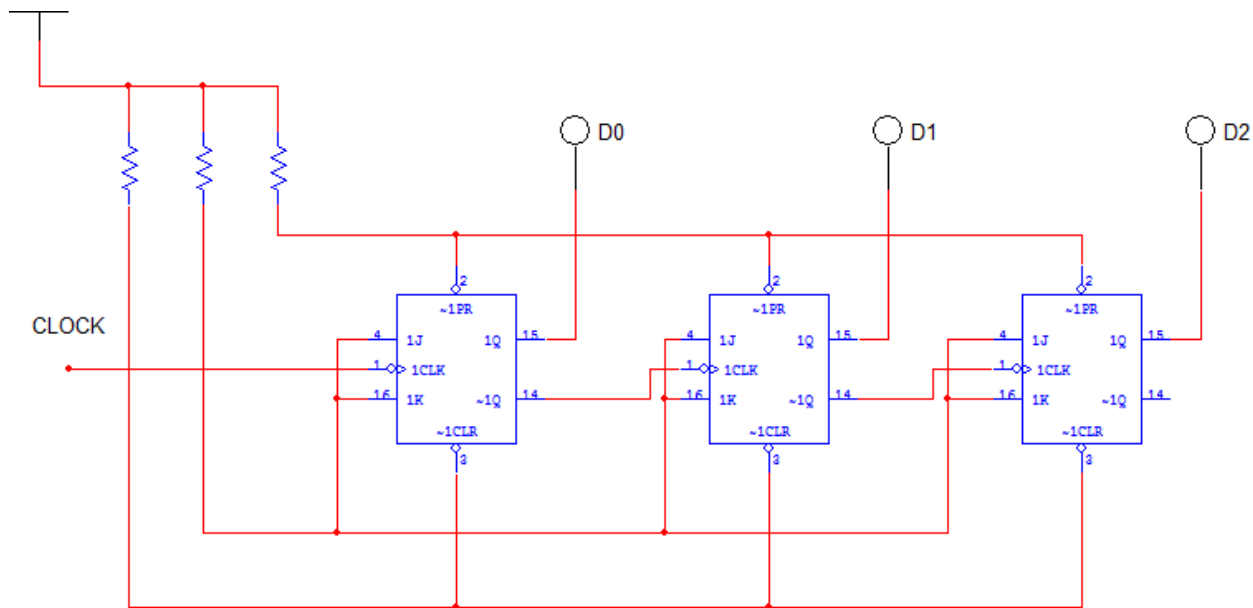
- $T_{p(\text{tot})} = 4 \times T_p = 4 \times 10 \text{ ns} = 40 \text{ ns}$

- A határfrekvencia:

- $f_{\text{max}} = 1 / T_{p(\text{tot})} = 1 / 40 \text{ ns} = 25 \text{ MHz}$

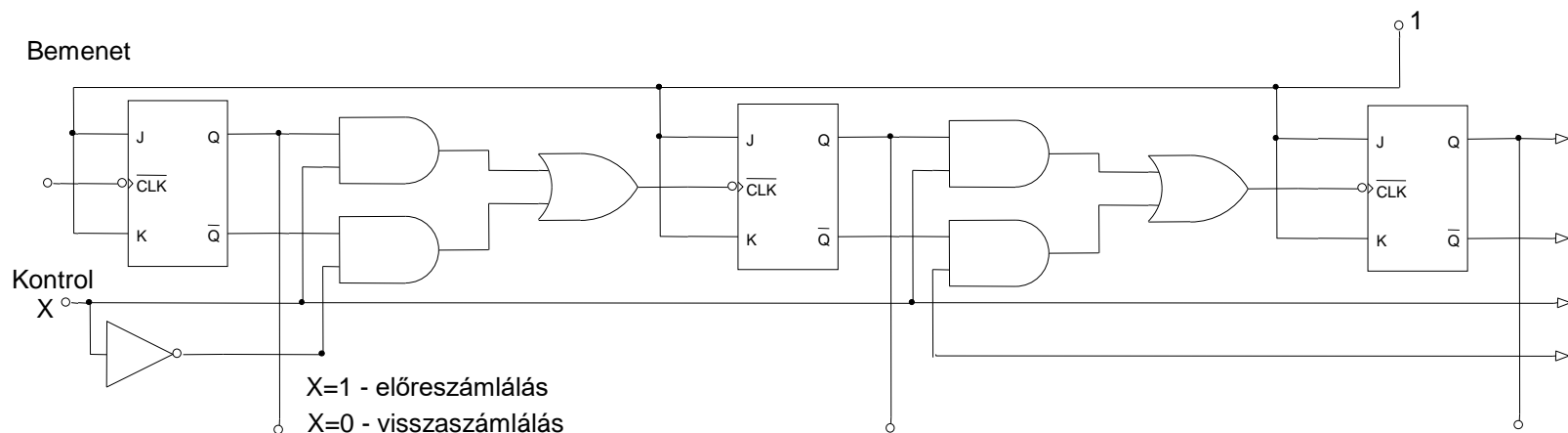
Aszinkron lefelé számlálók

- T tárolóknak a \overline{Q} kimenetét kötjük a következő tároló bemenetére.
- Lefutó élre vezérelt tárolókat alkalmazunk



Reverzibilis számlálók

- T tárolóknak a Q kimenetét kötjük a következő tároló bemenetére felfele számol
- T tárolóknak a \bar{Q} kimenetét kötjük a következő tároló bemenetére lefele számol
- Irányváltó számlálót úgy kapunk, egy kontrol jel segítségével (X) logikai kapukkal megvalósítjuk, ha X igaz, a Q kimenetek kerülnek a következő órajelbemenetre, ha X hamis, akkor meg a \bar{Q} kimenetek.

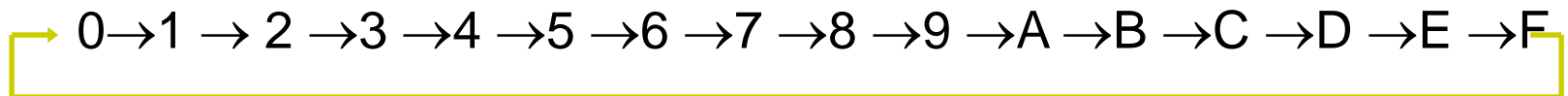


Számlálók ciklusának rövidítése

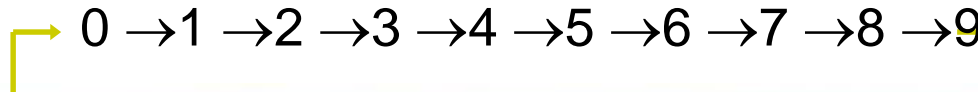
- Modulo-N számlálók építése, $N \neq 2$ -hatvány
- Az N-ik impulzus hatására a kezdő állapotba tér vissza a számláló
- Egy modulo-N számláló logikai tervezésének lépései:
 - a tárolók számának meghatározása (n),
$$2^{n-1} \leq N \leq 2^n;$$
 - egy n bites számláló logikai vázlat megrajzolása, olyan tárolókkal amelyeknek aszinkron törlő bemenete is van
 - a törlő bemeneteket olyan logikai hálózattal egészítünk ki, ami az N-ik impulzus hatására minden FF -ot töröl és ezzel az alap állapot áll be.

Pl. 4 bites számlálók ciklusai

BINÁRIS:



DECIMÁLIS:



Aszinkron decimális számláló

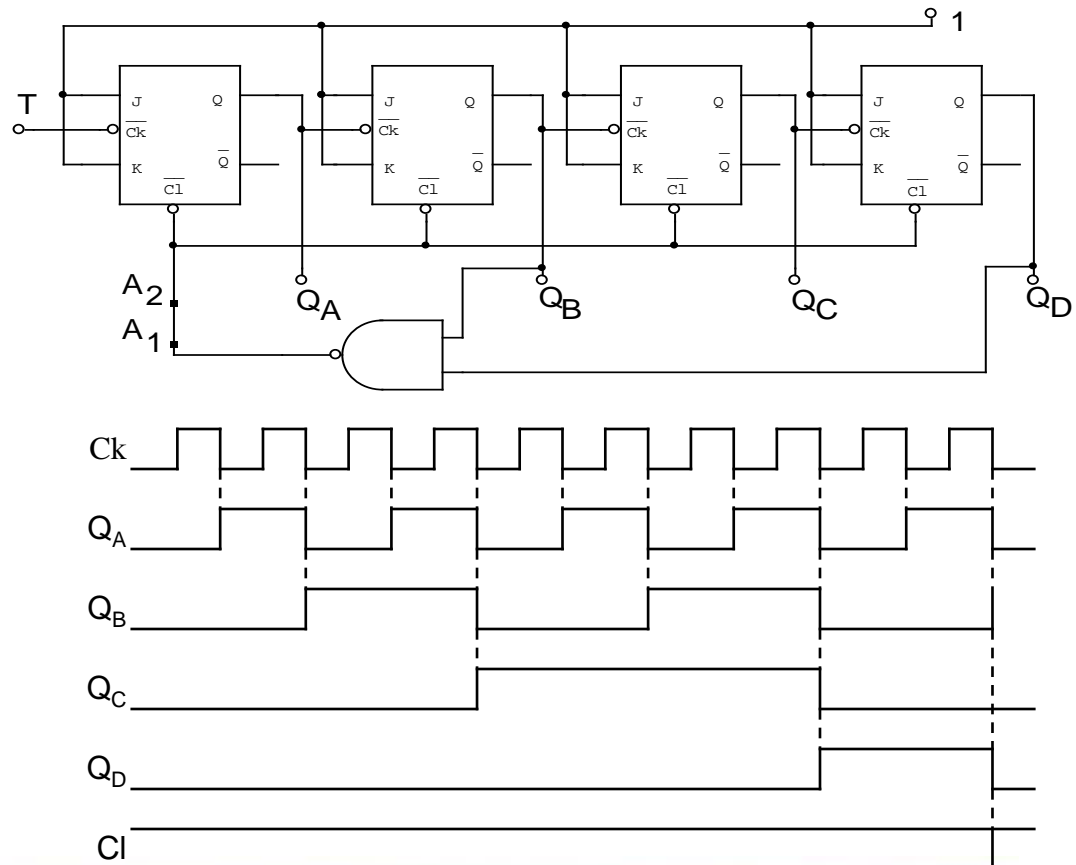
$$2^{n-1} \leq N \leq 2^n$$

$$N=10 \Rightarrow n=4$$

$$2^{4-1} \leq 10 \leq 2^4;$$

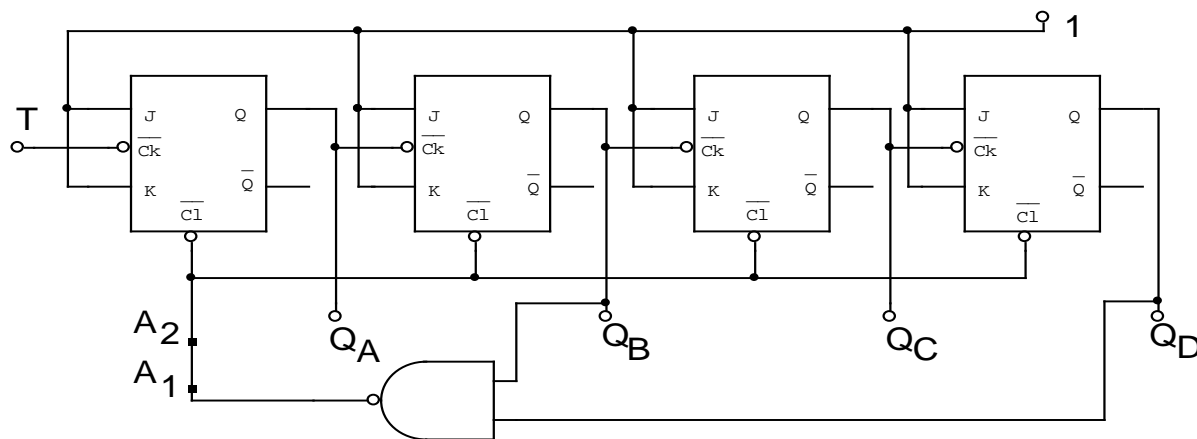
- egy 4 bites bináris számlálót - amelynek aszinkron törlő bemenete is van -
- olyan logikai hálózattal egészítünk ki, ami az 1 0 1 0 (decimális 10) állapot megjelenésekor minden FF-t töröl és ezzel 0 0 0 0 állapot áll be.

Imp. sz.	Q_D	Q_C	Q_B	Q_A
(Állapot) 0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 (0)	1(0)	0	1(0)	0

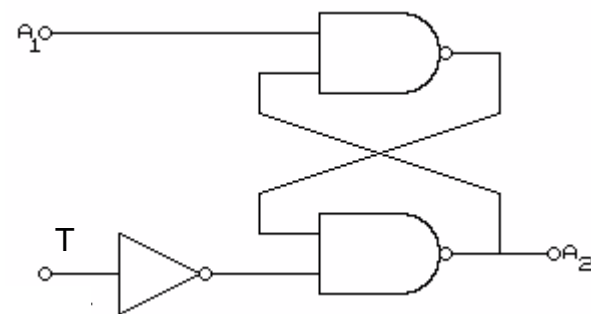


Aszinkron decimális számlálók (2)

- Ha a tárolók késleltetési ideje a Cl bemenetekhez képest nagyon eltérő egyes tárolók hamarabb törlődhetnek és a törlő feltétel eltűnik mielőtt minden tároló törlődött volna

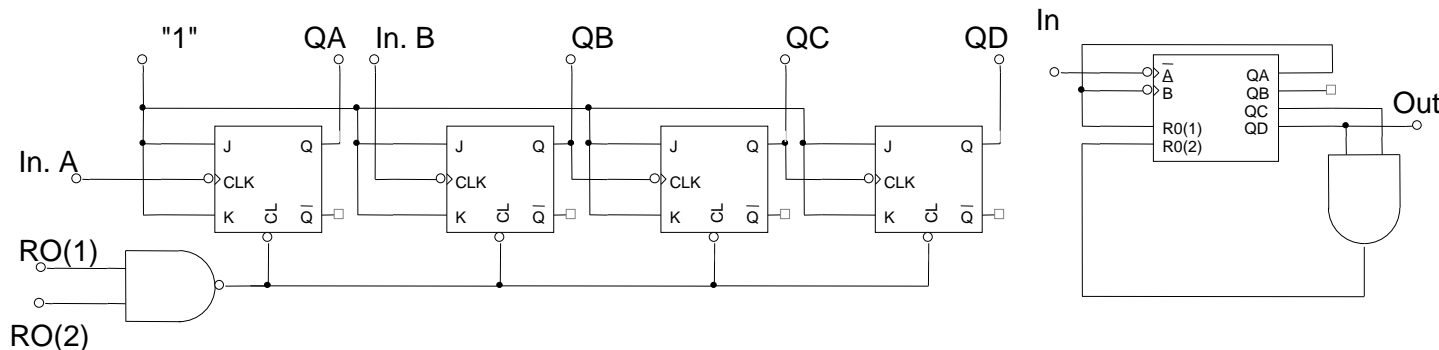


- A₁ és A₂ pontok köze beiktatunk egy alapít tárolót



Integrált áramköri aszinkron számlálók

- Aszinkron működésű számlálók
 - SN7490 – BCD számláló. Egy kettes és egy ötös osztóból áll.
Az R_{0i} jelű bemenetre adott 1 szinttel mind a négy flip-flop 0-ba állítható (törlés).
Az R_{0j} jelű bemenetek vezérlésével a BCD 1 0 0 1 kódot (9 beírása) tárolja a számláló.
 - SN7493 - 4 bites bináris számláló.
Ha In.A - ra adjuk a számlálandó jelet és az Q_A kimenetet In.B - vel kötjük össze.
 - Amennyiben az impulzusokkal az In.B-t vezéreljük és Q_D -t a In.A - val kötjük össze, olyan tízes osztót kapunk, amelynél az Q_A kimeneten szimmetrikus négyszögjelet kapunk.



- Milyen modulusú számlálót ábrázol a jobboldali ábra?

$$CI = R_{01} \& R_{02} = Q_A \& (Q_C \& Q_D)$$
$$M = 13$$

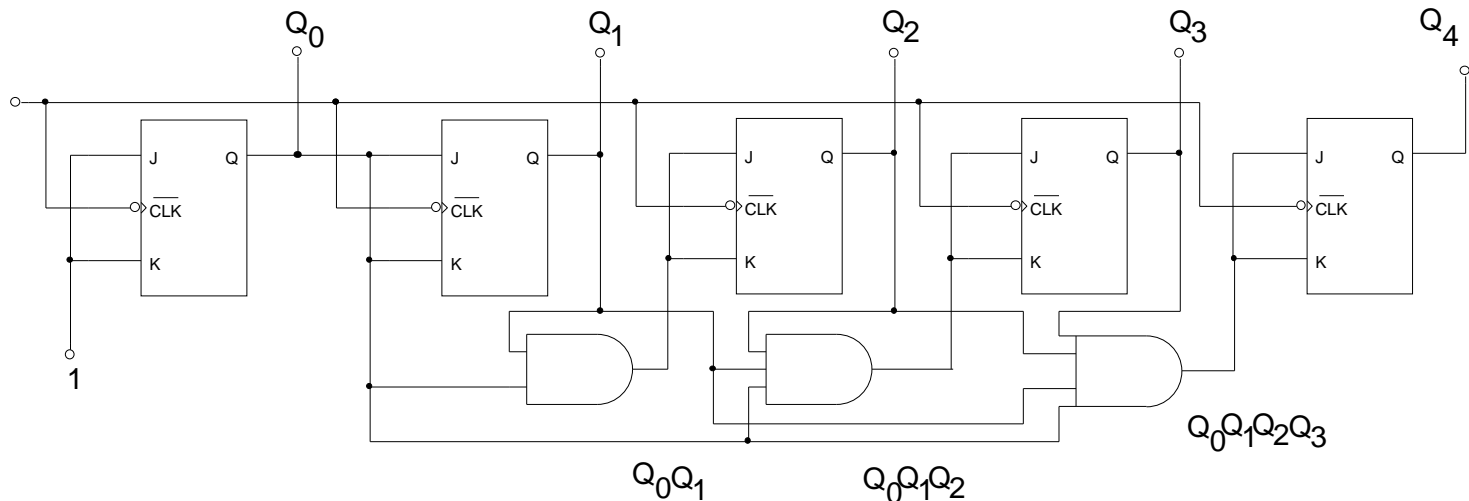
Szinkron bináris számlálók

- Az aszinkron számlálók hibáit kiküszöbölik.
- A külső órajel az összes flip-flopot egyszerre vezérli
- Q_0 minden órajelnél változtatja az állapotát, ezt úgy érjük el ha:
 - $T_0=J_0=K_0=1$
- A FF1 csak akkor vált ha $Q_0=1$, Ennek megfelelően
 - $T_1=J_1=K_1= Q_0$ (a bemeneteket a Q_0 - hoz kel kötni)
- A FF2 csak akkor vált ha $Q_0=Q_1=1$, Ennek megfelelően
 - $T_2=Q_0Q_1$
 - $T_3=Q_0Q_1Q_2$
 - $T_4=Q_0Q_1Q_2Q_3$

Nr.	Q_0	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0	0
1	1	0	0	0	0
2	0	1	0	0	0
3	1	1	0	0	0
4	0	0	1	0	0
5	1	0	1	0	0
6	0	1	1	0	0
7	1	1	1	0	0
8	0	0	0	1	0
9	1	0	0	1	0
10	0	1	0	1	0
11	1	1	0	1	0
12	0	0	1	1	0
13	1	0	1	1	0
14	0	1	1	1	0
15	1	1	1	1	0
16	0	0	0	0	1
17	1	0	0	0	1

Párhuzamos átvitelű szinkron bináris számlálók

$$\begin{aligned}T_0 &= 1 \\T_1 &= Q_0 \\T_2 &= Q_0 Q_1 \\T_3 &= Q_0 Q_1 Q_2 \\T_4 &= Q_0 Q_1 Q_2 Q_3\end{aligned}$$

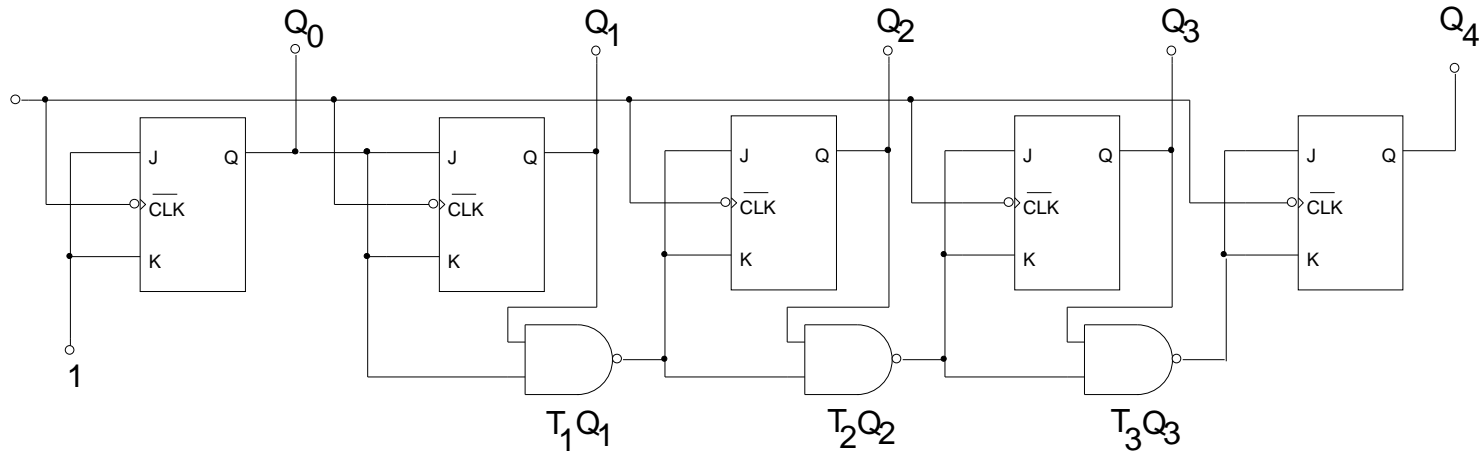


$$T_{\min} = t_{pb} + t_{pp}$$

az n -ik szinten $n-1$ bemenetű ÉS kapu kell

Soros átvitelű szinkron bináris számlálók

$$\begin{aligned}T_1 &= Q_0 \\T_2 &= Q_0 Q_1 = T_1 Q_1 \\T_3 &= Q_0 Q_1 Q_2 = T_2 Q_2; \\T_4 &= Q_0 Q_1 Q_2 Q_3 = T_3 Q_3\end{aligned}$$



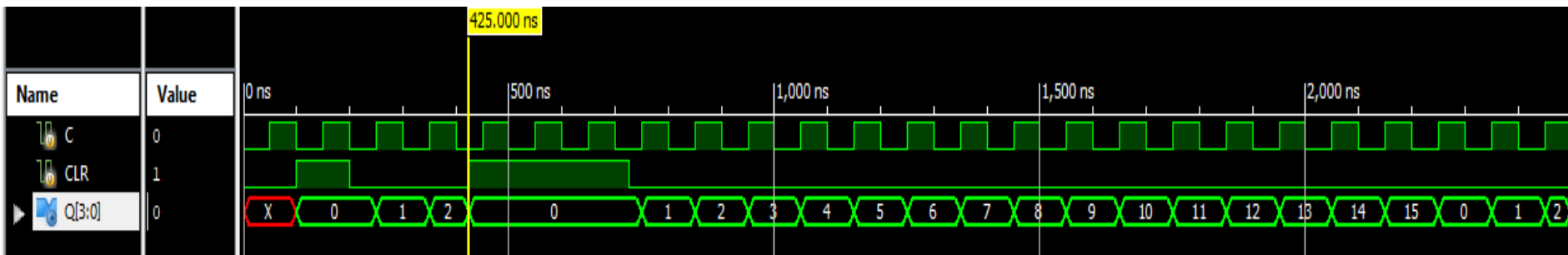
- soros átvitel hátránya - az ÉS kapukon keresztüli jelterjedést végig kell várjunk

$$T_{\min} = t_{pb} + (N-2) t_{pp}$$

Felfelé számláló Verilog tervezése

Aszinkron törlés

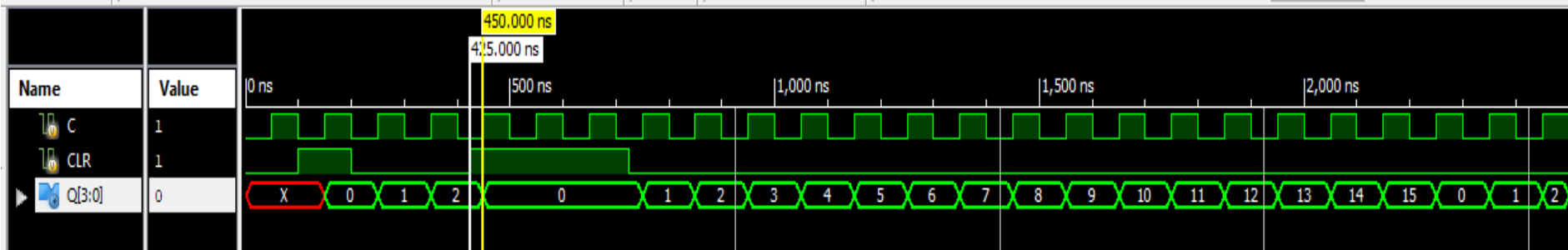
```
//  
// 4-bit up counter with an asynchronous clear.  
//  
module counter_1 (input C, CLR, output reg [3:0] Q);  
  
    always @(posedge C or posedge CLR)  
    begin  
        if (CLR)  
            Q <= 4'b0000;  
        else  
            Q <= Q + 1'b1;  
        end  
    endmodule
```



Felfelé számláló Verilog tervezése

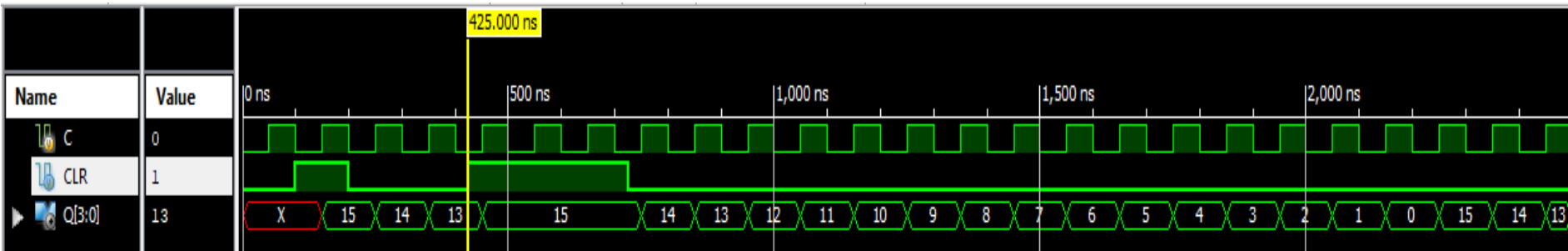
Szinkron törlés

```
//  
// 4-bit up counter with an synchronous clear.  
//  
module counter_2 (input C, CLR, output reg [3:0] Q);  
  
    always @(posedge C)  
    begin  
        if (CLR)  
            Q <= 4'b0000;  
        else  
            Q <= Q + 1'b1;  
        end  
    end  
endmodule
```



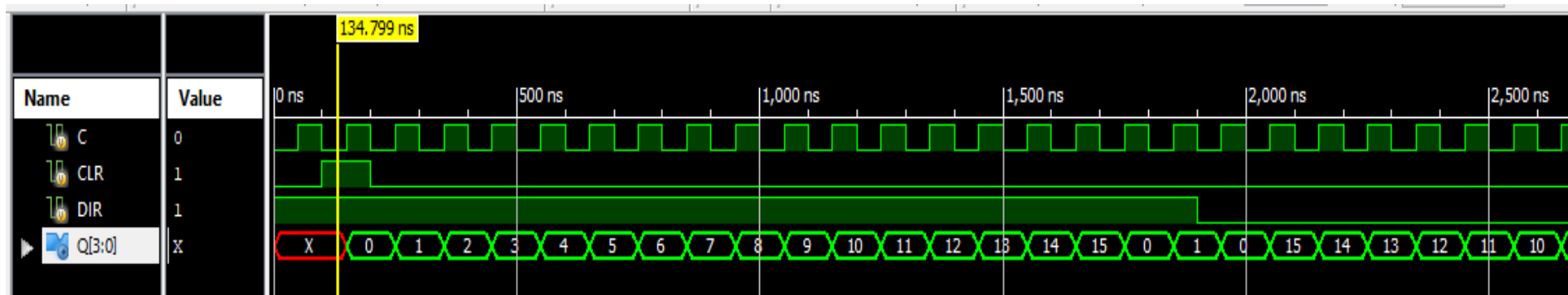
Lefelé számláló Verilog tervezése

```
module counter_3 (input C, CLR, output reg [3:0] Q);  
    always @(posedge C)  
    begin  
        if (CLR)  
            Q <= 4'b1111;  
        else  
            Q <= Q - 1'b1;  
        end  
    endmodule
```



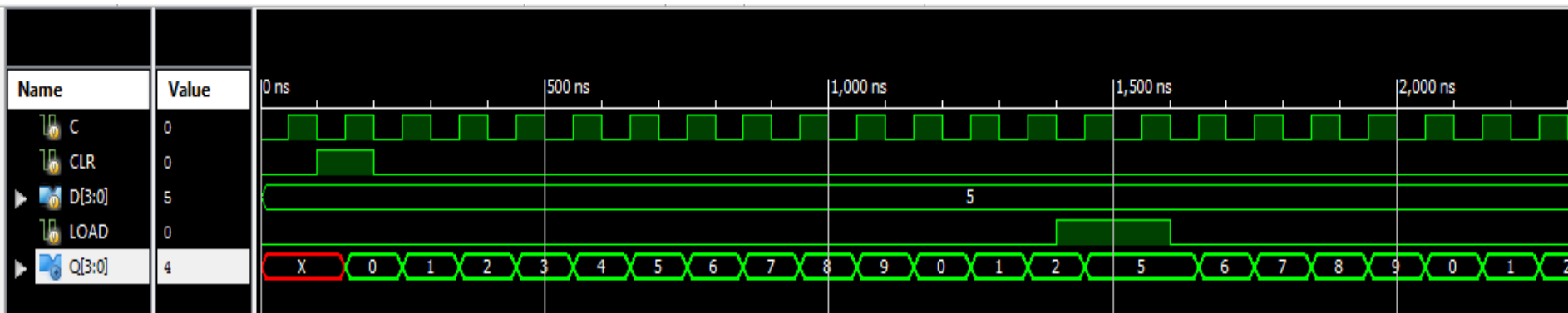
Fel/Le számláló Verilog tervezése

```
module counter_3 (input C, CLR, DIR, output reg [3:0] Q);
    always @(posedge C)
    begin
        if (CLR)
            Q <= 4'b1111;
        else if (DIR)
            Q <= Q + 1'b1;
        else
            Q <= Q - 1'b1;
    end
endmodule
```



Felfelé számláló, ciklus rövidítéssel és tölthető kezdő értékekkel

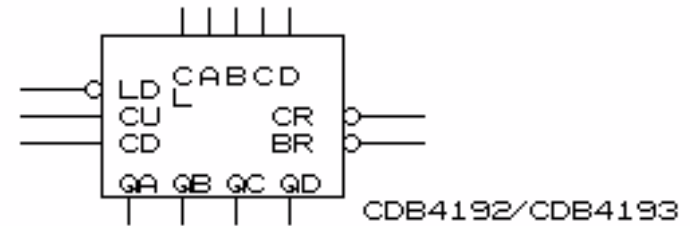
```
module counter_5 (input C, CLR, LOAD, input [3:0] D, output reg [3:0] Q);  
  
assign q9 = (Q== 4'd9); //assign q12 = (Q== 4'd12);  
  
always @(posedge C)  
begin  
    if (CLR | q9)  
        Q <= 4'b0000;  
    else if (LOAD) // (LOAD==1)  
        Q <= D; // vagy konstanssal Q <= 4'b0101;  
    else  
        Q <= Q + 1'b1;  
    end  
endmodule
```



Integrált áramköri szinkron számlálók

- Szinkron működésű számlálók

- SN74192 – reverzibilis BCD számláló.
- SN74193 - reverzibilis 4 bites bináris számláló.



- Q_A, Q_B, Q_C, Q_D - kimenetek.

- CR (carry) alacsony állapotával (0) jelzi a 1001, vagy 1111 kombinációt a kimenteken ($Q_D Q_C Q_B Q_A$) az 74192E, vagy 74193E és ha felfele számolunk.

- BR (borrow) alacsony állapotával (0) jelzi a 0000 kombinációt a kimenteken, és ha lefele számolunk.

- LD (load - betöltés) mikor = 0, a DCBA bementeken lévő értékeket tölti be

- CL (clear) törli a számalálókat.

- CU (count up – számolás felfele) CD egyesén tarjuk.

- CD (count down - számolás lefele).

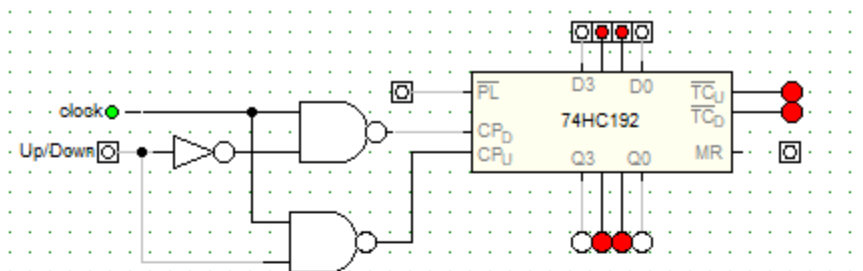
- D,C,B,A – adat bemenet.

- CR și BR több számláló összekötését szolgálják

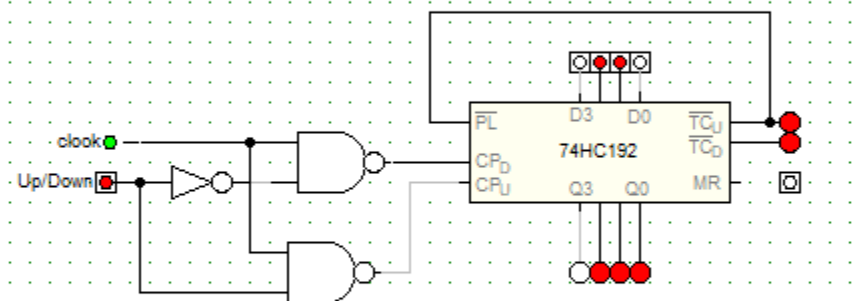
Integrált áramköri szinkron számlálók

SN74192 – reverzibilis BCD számláló.

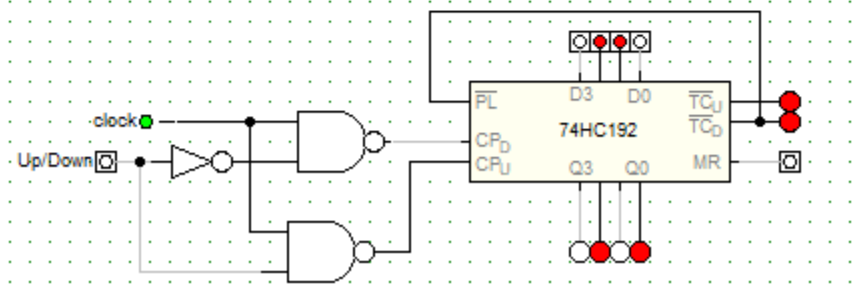
Töltés



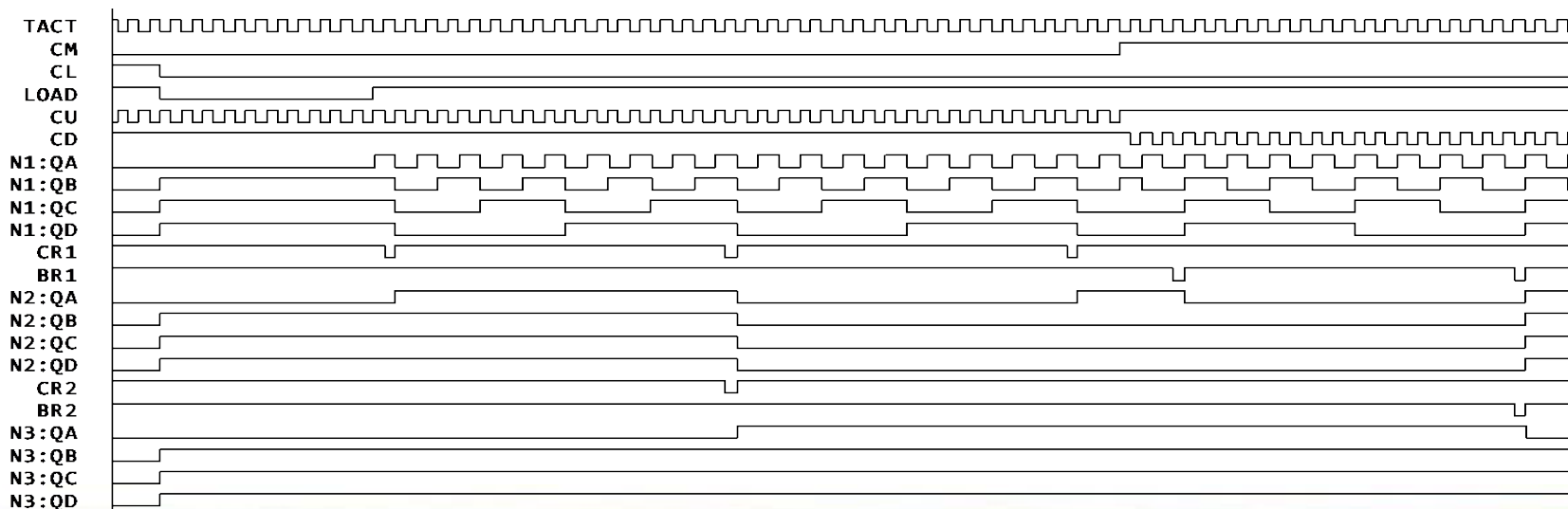
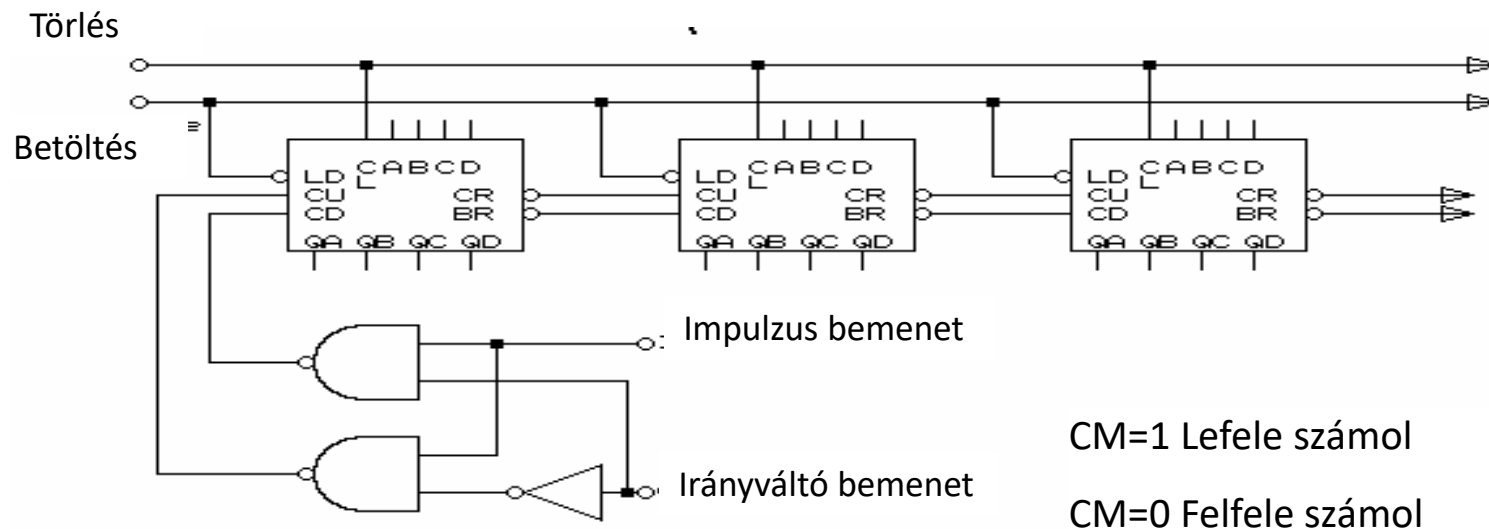
Számolás felfelé
(egy órajel után)



Számolás lefelé
(egy órajel után)



Szinkron számlálók Modulus növelés



Szinkron számlálók alkalmazási példa

