

4.4. Regiszterek

A számítógépek és egyéb digitális berendezések működésének elengedhetetlen feltétele az információ – átmeneti vagy hosszabb ideig tartó – tárolása, és az információhoz való gyors hozzáférés. A tárolók fontos jellemzőjének tekinthető a tárolt információ mennyisége és a hozzáférési idő. A nagy tárolási kapacitás és a rövid hozzáférési idő általában ellentmondó követelmények. A különböző tároló áramkörökkel részletesebben majd a 6. fejezetben fogunk foglalkozni.

Abban az esetben, amikor csak kevés információ tárolására (100 bites nagyságrendig) és gyors hozzáférésre van szükség (10÷20 ns), akkor a feladat megoldására flip-flopos tárolók alkalmazhatók. Az ilyen flip-flopos tárolókat – amelyeket általában csoportosan használnak egy nagyobb információ egység (pl. byte) tárolására – **regisztereknek** nevezzük.

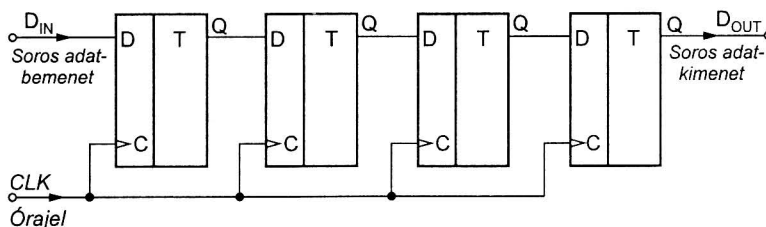
A regiszterek az információbevitel (beírás) és az információkiolvasás szempontjából *soros* vagy *párhuzamos hozzáférésűek* lehetnek. Soros beírásnál és kiolvasásnál a regiszter első és utolsó flip-flopjához lehet hozzáférni. Ebben az esetben szükséges, hogy az információt a regiszterben léptetni lehessen. Ezek a regiszterek a **léptetőregiszterek**.

Párhuzamos beírásnál és kiolvasásnál az információt a regiszter minden flip-flopjába egyszerre írják be, illetve onnan egyszerre olvassák ki. Mivel ezeknél a regisztereknél léptetés nem szükséges, a regiszter csak tárolási feladatra alkalmas. Ezeket a típusokat **átmeneti tároló** vagy **közbenső tároló (puffer) regisztereknek** nevezik.

4.4.1. Léptetőregiszterek

A léptetőregiszter flip-flopok olyan láncja, amely lehetővé teszi, hogy a bemenetére adott információ minden egyes órajel hatására egy flip-floppal tovább lépjen. A léptetés irányának a megadására a *jobbra*, illetve *balra* megjelöléseket alkalmazzák. A bemeneti jel áthaladva a láncon késleltetve, de egyébként változatlanul jelenik meg a kimeneten. A léptetőregiszter bináris jelek tárolására szolgál és minden egyes flip-flop, amely a regisztert alkotja egy bit információ tárolására alkalmas. A léptetőregiszterekben a megfelelő működés érdekében (minden léptetési parancsra egy és csakis egy léptetés) feltétlenül órajelvezérelt flip-flopokat kell alkalmazni.

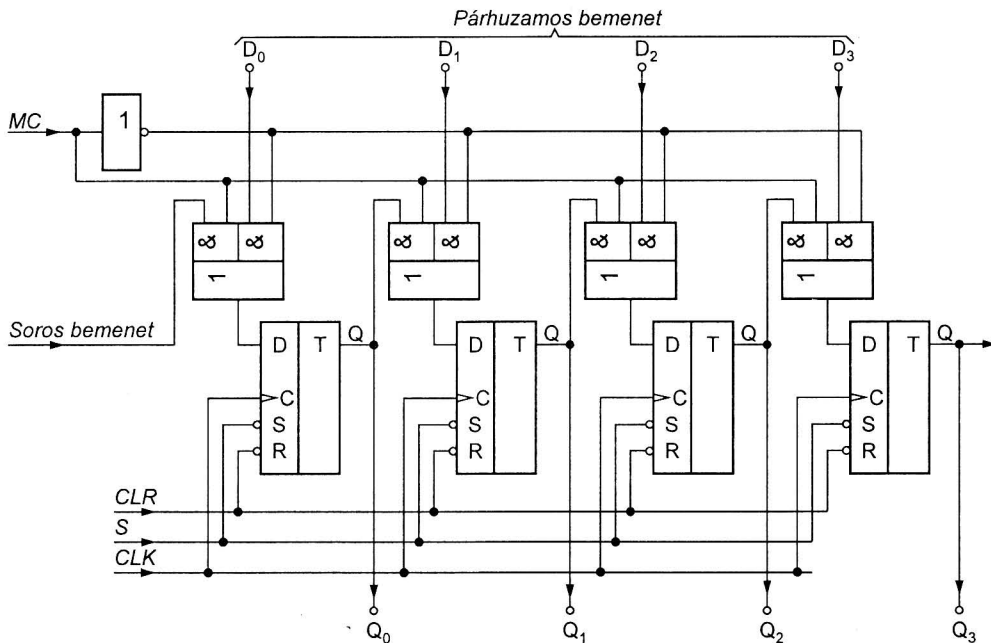
A léptetőregiszterek esetén a soros és párhuzamos beírás és kiolvasás, valamint a kétféle léptetési irány lehetőségének variációival sokféle típus állítható elő. A 4.59. ábra – a legegyszerűbb változatnak tekinthető – egyirányban (*jobbra*) léptethető, soros beírású és kiolvasású 4-bites áramkör kapcsolását szemlélteti.



4.59. ábra. 4-bites soros léptetőregiszter

Ha a 4.59. ábrán látható áramkör minden flip-flopját kimeneti kivezetéssel látjuk el, a léptetőregiszter párhuzamos kiolvasásra is alkalmassá válik.

A flip-flopok statikus beíró és törlő bemeneteit is felhasználva lehetővé válik, hogy a léptetőregiszter párhuzamos beírásra is alkalmassá váljon. A 4.60. ábra egy 4-bites soros/párhuzamos kiolvasású és beírású léptetőregiszter kapcsolási rajzát szemlélteti. A párhuzamos beírás és a léptetés funkcióját az ÉS-VAGY elemek választják szét. A beírás előtt szükséges törlés az *CLR* (törlés) bemeneten keresztül történik logikai **0** szinttel. Beírni az az *S* bemenetről logikai **0** szinttel lehet.



4.60. ábra. 4-bites soros/párhuzamos kiolvasású és beírású léptetőregiszter

MC (Mode Control) – üzemmód vezérlő

CLR (Clear) – törlés

S (Set) – beírás

CLK (Clock) – órajel

Ha az üzemmódvezérlő (angolul: *MC*– *Mode Control*) bemenetén logikai **0** szint van, a kettős ÉS kapucsoportok kapui közül azok vannak engedélyezve, amelyek a párhuzamos bemenetről (D_0, \dots, D_3) veszik az információt. Ha a vezérlőbemenet logikai **1**-es szinten van, az ÉS kapucsoportok kapui közül azok vannak engedélyezve, amelyek a szomszédos flip-flop kimenetére csatlakoznak. Így az $MC = 0$ párhuzamos beírás, az $MC = 1$ léptetés, illetve a soros bemenet felhasználásával soros beírás üzemmódot biztosít. Természetesen bármelyik működés csak órajel hatására jön létre.

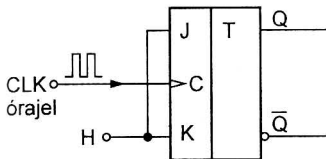
4.5. Számláló áramkörök

A szekvenciális hálózatok egyik igen fontos csoportjának tekinthetők a számláló áramkörök. A számláló áramkörök feladata, hogy rögzítsék és jelezzék (számlálják) a bemenetükre jutó impulzusok számát. Elvileg számláló áramkör lehet minden olyan áramkör, amelynél – bizonyos korlátokon belül – a beérkezett impulzusok száma és a kimeneti változók állapota között egyértelmű kapcsolat van.

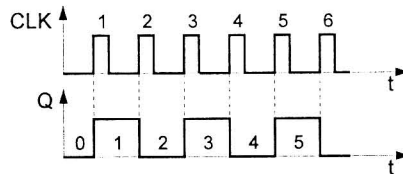
A számláló áramköröknek tárolniuk kell a beérkezett impulzusok számlálásának az eredményét és ezt újabb impulzus hatására megfelelő módon meg kell változtatniuk. Egy számláló áramkörnek legalább annyi egymástól különböző állapottal kell rendelkeznie, amennyi a számlálandó impulzusok számának a maximuma.

A számláló áramkörök legtipikusabb alapáramkörének a T flip-flop tekinthető. A J-K flip-flopok alkalmazása alapáramkörként nagyobb lehetőséget nyújt különleges logikai függvénykapcsolatok megvalósítására. Minden bit tárolására egy flip-flop szükséges. Egy négybites számláló (amely tehát 4 flip-floppal rendelkezik) maximálisan $N = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 15$ -ig számlálhat.

A 4.61. ábra egy T flip-floppal kivitelezett 1-bites számláló kapcsolását és a számlálás idődiagramját szemlélteti. Megfigyelhető, hogy a T flip-flop minden órajelre ellenkező állapotba billen át. Mivel csupán két megkülönböztethető állapota van, csak két órajelimpulzus egyértelmű leszámolására alkalmas. Kettőnél több impulzus leszámolásához természetesen kettőnél több különböző állapot szükséges, ami a számlálóban alkalmazott flip-flopok számának növelésével érhető el. Mivel minden kimeneti változó csak két értéket vehet fel, ezért n kimenet esetén 2^n kombináció fordulhat elő (vagyis a számláló 2^n darab impulzus leszámolására képes). Gyakran a lehetséges kombinációk csak egy részét használják fel.



a) kapcsolása



b) kimeneti állapotainak időábrája

4.61. ábra. 1-bites számláló

A számláló áramkörök nagyon sok típusa ismeretes, ezért osztályozásuk igen sokféle szempont szerint történhet. A következőkben a rendszerezés kedvéért megemlítjük a legfontosabb szempontokat.

- ◆ A számlálót alkotó flip-flopok órajellel való vezérlése szempontjából, megkülönböztetünk:
 - **aszinkron számlálót:** az órajelek a számlálónak általában csak az egyik, többnyire a legkisebb helyiértéket képviselő flip-flopját vezérlik. A többi flip-flop az órajel egyrástól kapja, így billenésük nem azonos időpontban történik (vagyis a működés aszinkron);
 - **szinkron számlálót:** az órajelek a számlálót alkotó összes flip-flopot egyszerre vezérlik. Ebben az esetben minden flip-flop billenése azonos időpontban történik (vagyis a működés szinkron).

- ◆ A számláló számábrázolása (a tárolt impulzusszám kódolása) szerint, megkülönböztetünk:
 - **bináris számlálót:** a tárolt impulzusszám kódolása bináris számrendszerben történik;
 - **decimális számlálót:** a tárolt impulzusszám kódolása (általában 8421 kódú) decimális számrendszerben történik.
- ◆ A számlálás iránya (sorrendje) szerint, megkülönböztetünk:
 - **előre-számlálót:** a tárolt impulzusszám növekvő sorrendű (angol elnevezése: up counter);
 - **vissza-számlálót:** a tárolt impulzusszám csökkenő sorrendű (angol elnevezése: down counter);
 - **reverzibilis-számlálót:** a számlálás iránya megfordítható (angol elnevezése: up-down counter).

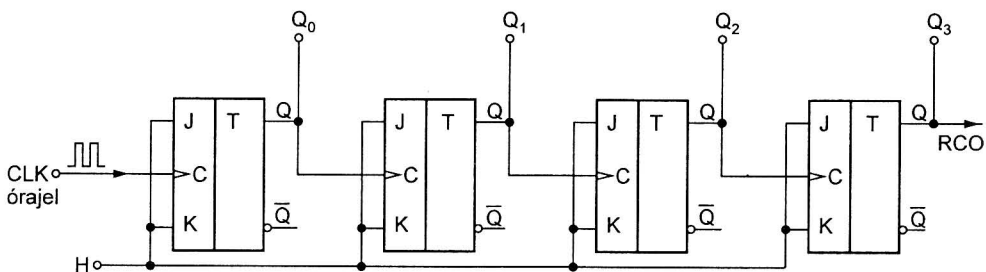
A csak csökkenő sorrendű számlálók áramköri megvalósítása (integrált áramköri formában) nem szokásos, általában a visszaszámlálás megvalósítására reverzibilis számlálókat alkalmaznak.

4.5.1. Bináris számláló áramkörök

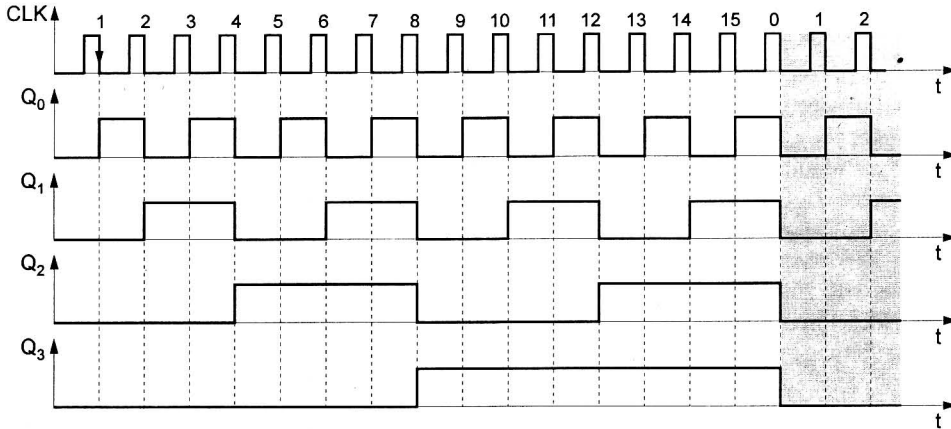
A számláló áramkörök legegyszerűbb felépítésű változatai a bináris számlálók. Jellegzetességük, hogy a tárolt impulzusszám kódolása bináris számrendszerben történik

4.5.1.1. Aszinkron bináris számlálók

Aszinkron bináris számláló áramkört kapunk, ha a 4.62. ábrán látható módon több J-K tárolót sorbakötünk, és ezek órajel-bemenetét mindig az előző tároló Q kimenetéhez csatlakoztatjuk. Az előreszámlálás elérésére negatív élvezérlésű tárolót alkalmazunk ($J=K=1$ feltétellel), így a tároló kimeneti állapota akkor változik mikor az órajel 1-ről 0-ra vált ($H-L$ átmenet esetén). A számláló tetszés szerint bővíthető tovább az **RCO** (Ripple Carry Output – átvitelkimenet) segítségével (pl. 10 tárolóval már 1023-ig számlálhatunk).



4.62. ábra. 4-bites aszinkron bináris számláló



4.63. ábra. Aszinkron bináris számláló kimeneti állapotainak időábrája

N	Q_3	Q_2	Q_1	Q_0
	2^3	2^2	2^1	2^0
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

4.8. táblázat. A kimenetek állapota az impulzusszám függvényében

(visszaszámláló). Egy ilyen visszaszámláló törvényszerűségei a 4.8. táblázatból kiolvashatók, ha a sorokat alulról felfelé tekintjük át.

1. Egy Q_i kimeneti változó visszaszámlálásnál mindig akkor változtatja meg az értékét, ha a következő kisebb helyértékű változó (Q_{i-1}) logikai 0-ról 1-re vált.
2. Egy visszaszámláló esetében a Q_i kimeneti változó mindig akkor változtatja meg az értékét, ha az összes kisebb helyértékű változó (Q_{i-1}, \dots, Q_0) logikai 0 értékű, és egy újabb impulzus érkezik.

A 4.63. ábrán látható időábra a számlálандó impulzusok (órajel) és az alkalmazott négy flip-flop kimeneteinek állapotai közötti összefüggést szemlélteti. A 4.8. táblázat a beérkezett impulzusok N száma és a Q_i kimeneti változók értékeinek összerendelését mutatja (természetesen ezek az összefüggések a 4.63. ábrán látható idődiagramból is kiolvashatók). A táblázatot felülről lefelé vizsgálva két törvényszerűség ismerhető fel:

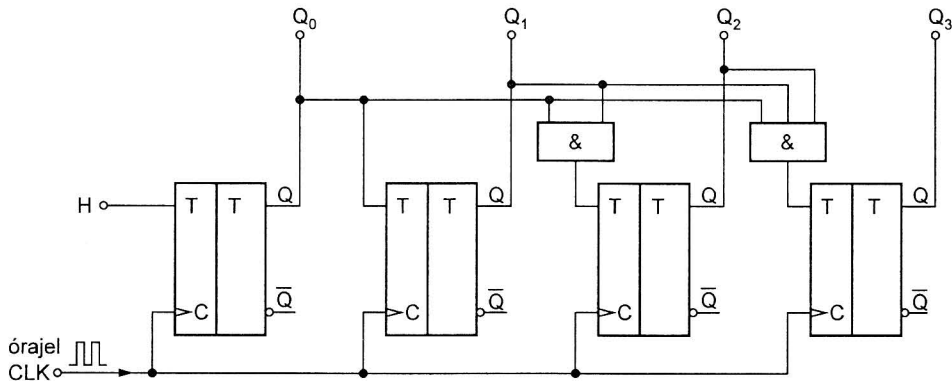
1. egy Q_i kimeneti változó akkor változtatja meg az értékét, ha az előtte álló kisebb helyértékű változó (Q_{i-1}) logikai 1-ről 0-ra vált;
2. egy Q_i kimeneti változó mindig akkor változtatja meg az értékét, ha minden kisebb helyértékű változó (Q_{i-1}, \dots, Q_0) logikai 1 értékű, és egy újabb impulzus érkezik.

Vannak esetek mikor olyan számláló áramkörre van szükség, amelynél a számláló állását minden egyes számlálандó impulzus 1-gyel csökkenti

4.5.1.2. Szinkron bináris számlálók

Az aszinkron számlálók hátránya a gyors ismétlődésű impulzusok számlálásánál mutatkozik meg. Bizonyos szintet meghaladó sebességnél a számláló állapota hamis lehet. Ennek oka, hogy a nagyobb helyértékű flip-flop felé az előbbieket késleltetési ideje láncszerűen adódik össze.

A *szinkron számlálók* az előbbi hátrányt küszöbölik ki. A flip-flopok órajel bemenetei egymás között össze vannak kapcsolva, így a számlálandó impulzusok szinkronban vezérlik a számláló összes flip-flopját. A szinkron számlálóknál az állapotváltozást a tárolók (általában J-K flip-flop) logikai bemenetein alkalmazott vezérlés határozza meg. Bináris számlálás esetén a logikai feltételek igen egyszerűek. A 4.8. táblázatot tanulmányozva már megállapítottuk, hogy előreszámláláskor egy adott flip-flop akkor billen át, ha az összes kisebb helyértékű képviselő flip-flop logikai 1-et tárol. Ennek a feltételnek a teljesülését *ÉS* kapukkal lehet ellenőrizni. Egy 4-bites szinkron bináris előreszámláló kapcsolási rajzát a 4.64. ábra mutatja. A számláló pozitív élvezérelt *T* flip-flopokból épül fel. Az órajel ebben az esetben csak az állapotváltozások ütemét határozza meg.

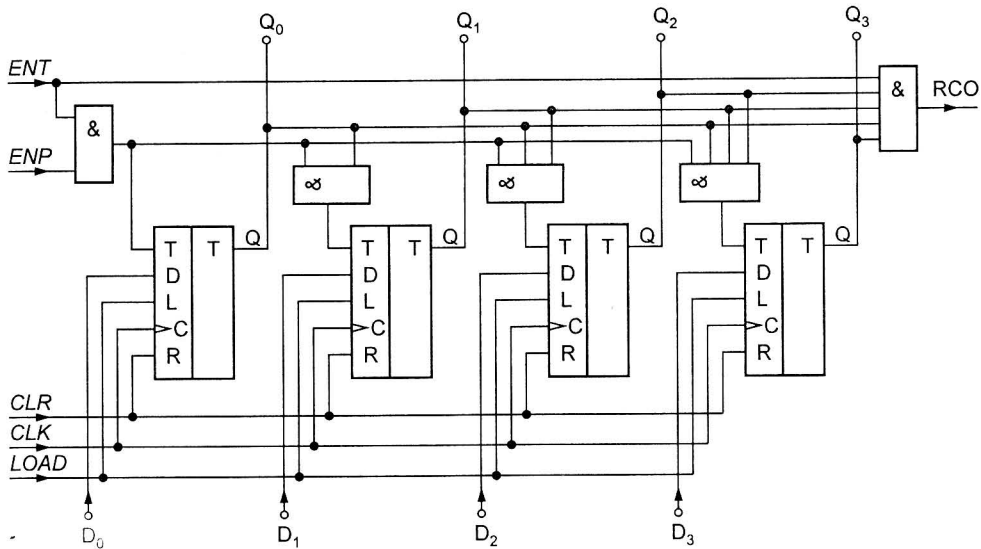


4.64. ábra. Szinkron bináris előreszámláló

Annak ellenére, hogy a bemeneti impulzusok egyidejűleg minden *C* óra jelbemenetre rákerülnek, mégsem változik minden órajel hatására az összes tároló állapota. A tárolók csak akkor billennek át, ha a *T* vezérlőváltozó értéke logikai 1-es.

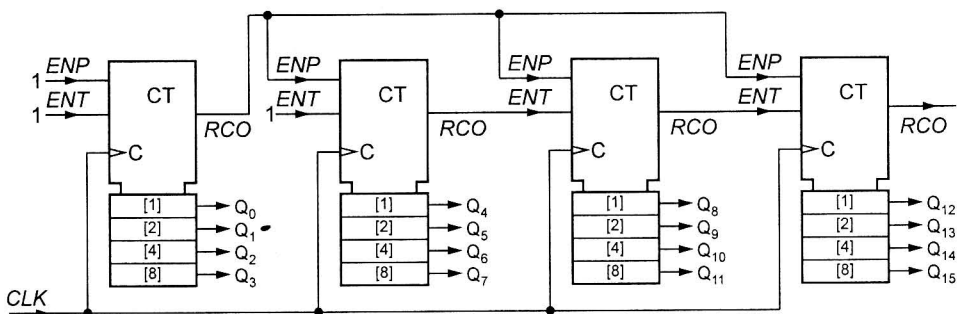
Az integrált szinkron számlálóknak még további be- és kimenetei vannak, amelyek szerepét és alkalmazását a 4.65. ábra alapján megérthetjük.

- A *CLR* (Clear) törlő bemenet alkalmas az egész számlánc nullázására.
- A *LOAD* beíró bemeneten keresztül vezérelve a számlálóba tetszőleges *D* szám írható be.
- Az *RCO* (Ripple Carry Output) átvitelkimenet segítségével többhelyértékű számlálót kapcsolhatunk össze. Az átvitelkimenetnek akkor kell logikai 1-nek lennie, ha a számláló állása eléri az 1111-et, és minden kisebb helyértékű tároló-egység ugyancsak átvitelt ad.
- Az *ENT* (Enable T) *T* engedélyező bemenet.
- Az *ENP* (Enable P) *P* engedélyező bemenet.



4.65. ábra. Integrált szinkron bináris előreszámláló kapcsolása

Többhelyértékű számlálót, pl. több 4-bites számláló kaszkád kapcsolásával alakíthatunk ki. A fokozatokat az *RCO* kimenetek és az *ENT* engedélyező bemenetek összekötésével kapcsolhatjuk össze. A sorbakapcsolt *ÉS* kapuk késleltetési ideje ebben az esetben összegeződik és emiatt csökkenne a maximális számlálási frekvencia. Ezért jobb megoldásnak tekinthető, ha a szükséges *ÉS*-függvénykapcsolatokat minden egyes számláló fokozatban párhuzamosan alakítjuk ki. E célból kihagyjuk a legkisebb helyértékű számláló fokozatot a soros *RCO-ENT* láncból és engedélyezzük párhuzamosan a nagyobb helyértékű számláló fokozatokat az *ENP* bemeneteken. Ily módon a 4.66. ábrán látható elrendezéssel a párhuzamos *ÉS*-függvényeket külső logikai kapuk nélkül valósíthatjuk meg.



4.66. ábra. Szinkron bináris előreszámláló fokozatok kaszkád kapcsolása

CT (Counter) – számláló

ENT (Enable T) – T engedélyező bemenet

ENP (Enable P) – P engedélyező bemenet

CLK (Clock) – órajel

RCO (Ripple Carry Output) – átvitelkimenet

4.5.1.3. Előre-hátra számlálók (reverzibilis számlálók)

Az *előre-hátra számlálók* (más néven *reverzibilis számlálók*) olyan számláló áramkörök, amelyeknél a számlálás iránya megfordítható. A reverzibilis számlálók esetén két típust lehet megkülönböztetni. Az egyiknél két bemenet van, az órajelbemenet és a másik, amely meghatározza a számlálás irányát. A másik típus esetén két órajelbemenet van, az egyikre kapcsolt jel növeli, a másikra pedig csökkenti a számláló által képviselt értéket.

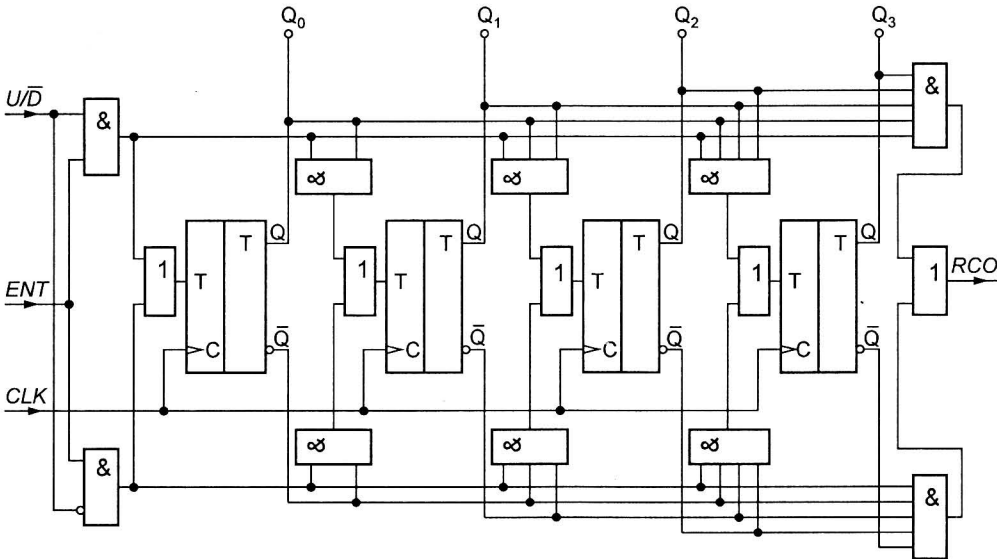
A 4.67. ábra egy átkapcsolható számlálási irányú számláló kapcsolási rajzát mutatja. A hátraszámlálás billenési feltétele, hogy minden egyes flip-flopnak csak akkor kell állapotot változtatnia, ha minden kisebb helyértékű helyen logikai 0 van. Ennek a feltételnek a dekódolása az ábrán látható kapcsolásban úgy történik, hogy kapuzásra a Q kimenetek helyett a \bar{Q} kimeneteket használjuk. A számlálás irányának az átkapcsolása az U/\bar{D} előre-hátra vezérlőjel segítségével történik, mely vagy a kapcsolás felső részének (*előreszámlálás*) vagy az alsó részének (*hátraszámlálás*) kapuhálózatát engedélyezi.

Abban az esetben ha a számlálót tovább szeretnénk bővíteni, akkor átvitelre a következő fokozat felé két esetben van szükség:

1. ha a számláló állása: **1111** ($U/\bar{D} = 1$, előreszámlálásnál).
2. ha a számláló állása: **0000** ($U/\bar{D} = 0$, hátraszámlálásnál).

Az átviteli változóra a következő logikai függvény adódik:

$$RCO = [Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot U/\bar{D} + \bar{Q}_0 \cdot \bar{Q}_1 \cdot \bar{Q}_2 \cdot \bar{Q}_3 \cdot \overline{U/\bar{D}}] \cdot ENT$$



4.67. ábra. Átkapcsolható számlálási irányú bináris számláló

4.5.2. Decimális számláló áramkörök

A bináris számlálók esetén megfigyelhető, hogy a három helyértékű változat 7-ig számlálhat, a négy helyértékű számláló pedig 15-ig. A természetes BCD ábrázolású számjegyeket négy helyértékű bináris szám képviseli, ahol a számjegyek súlya 2^3 , 2^2 , 2^1 , 2^0 , ezért a BCD ábrázolást 8421 kódnak is nevezik. Ennek megfelelően egy 8421 kódú BCD számláló megvalósításához négy bináris helyérték szükséges.

N	Q_3	Q_2	Q_1	Q_0
	2^3	2^2	2^1	2^0
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	0	0	0	0

4.9. táblázat. BCD kód állapotdiagramja

A 4.9. táblázat a 8421 típusú BCD kód állapotdiagramját szemlélteti (N a bemenő impulzusok számát képviseli). A táblázat a 9. számjegyig megegyezik a 4.8. táblázat szerinti bináris számlálókra vonatkozó táblázattal, a 10. számot viszont ismét a **0000** érték képviseli.

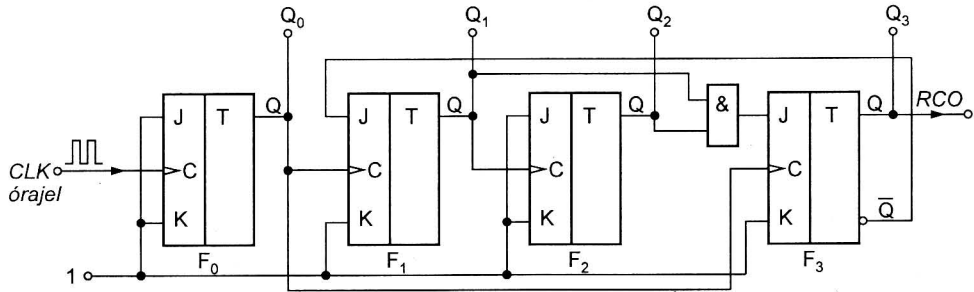
A decimális számláló csak abban különbözik a bináris számlálótól, hogy a tizedik impulzus hatására nullázódik, és átvitelt képez. A kapott átvittel a következő, eggyel nagyobb helyértékű decimális számláló áramkört vezérelhetünk. Ahhoz, hogy a számlálót a tizedik bemeneti impulzus kezdeti állapotába juttassa egy megfelelő kiegészítő logika is szükséges, amelyet kombinációs hálózattal könnyen meg lehet valósítani.

4.5.2.1. Aszinkron decimális számláló áramkörök

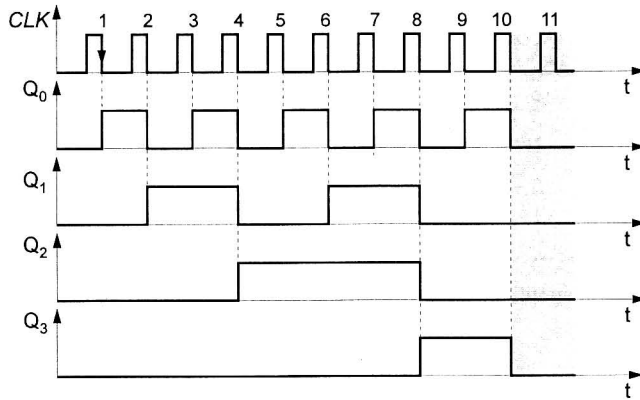
Decimális számláló áramkörök megvalósítására célszerű J - K flip-flopokat alkalmazni, mivel szélesebb körű vezérlési lehetőségeket biztosítanak. Aszinkron decimális számlálót kapunk a 4.60. ábrán látható bináris változat kapcsolásának módosításával, ha biztosítjuk a következő feltételek teljesülését:

- a Q_1 kimenet maradjon logikai **0** akkor, ha a Q_3 előzőleg már logikai **1** szintű lett (vagyis a Q_1 kimenetű tárolónak nem szabad átbillennie a tizedik impulzus hatására);
- a Q_3 kimenetnek a tizedik impulzus hatására logikai **1**-ről **0**-ra kell átváltania.

Megoldás: – a második F_1 jelű flip-flop J bemenete nem állandó logikai 1-es szinten van, hanem a negyedik F_3 jelű flip-flop negált kimenetéről kapja a logikai vezérlőjelét (4.68. ábra). Ennek következtében az F_1 jelű flip-flop J bemenete a számláló 0, ..., 7 állapotaiban logikai 1-es szinten van, a **nyolcadik** impulzus hatására ($Q_3 = 1$) logikai **0** szintre vált és az első három tároló törlődik. A következő **kilencedik** impulzus csak az első F_0 jelű tárolót billenti át, míg a **tizedik** impulzus a bemeneti logikai feltételek miatt csak az első flip-flopot billenti át $Q_0 = 1$ logikai állapotba (a Q_1 kimenet logikai **0**-ban marad). Mivel a Q_1 kimeneten nem volt állapotváltozás, a harmadik flip-flop nem kap órajelet ($Q_2 = 0$) és a negyedik flip-flop pedig törlődik ($Q_3 = 0$). Következésképpen a számláló a tizedik órajel hatására alapállapotba (**0000** logikai állapotba) kerül.



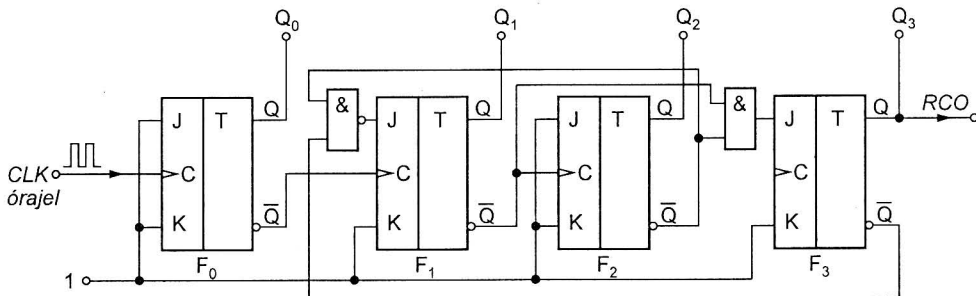
4.68. ábra. Aszinkron decimális számláló



4.69. ábra. Aszinkron decimális számláló kimeneteinek idődiagramja

Az aszinkron decimális előreszámláló kapcsolás kimeneti állapotainak időábráját a 6.69. ábra szemlélteti.

Aszinkron decimális visszaszámoló áramkör kapcsolását mutatja a 4.70. ábra. Az eltérés a 4.68. ábrán látható változattól abban nyilvánul meg, hogy ebben az esetben – a visszaszámolás követelményeinek megfelelően – a flip-flopok vezérlése a negált kimenetekről történik. A NAND kapuval előállított $J = 0$ logikai szint megakadályozza az F_1 jelű flip-flop átbillenését a 0→9 átmenetnél.



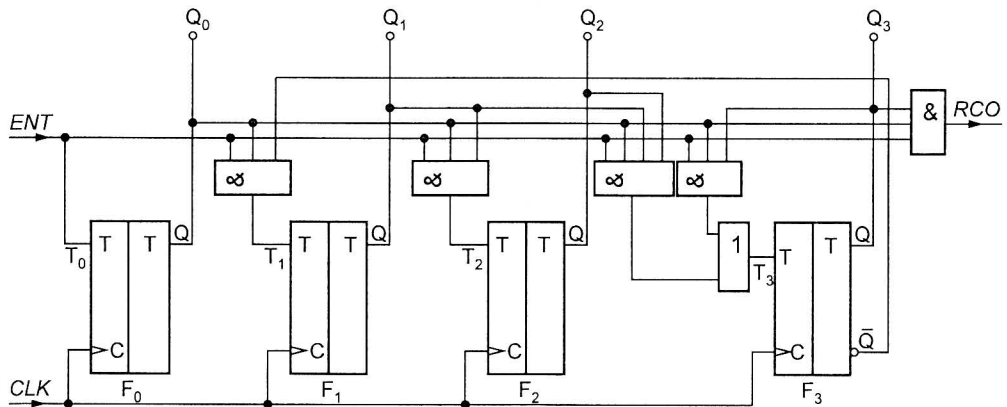
4.70. ábra. Aszinkron decimális visszaszámoló

4.5.2.2. Szinkron decimális számláló áramkörök

A 4.71. ábrán látható szinkron decimális előreszámláló áramkör a 4.64. ábrán látható bináris változat megfelelője a következő kiegészítésekkel:

- az F_3 jelű tároló negált kimenete vezérli az F_1 jelű tároló T bemenetét (a megfelelő logikai hálózaton keresztül), amely a $9 \rightarrow 0$ átmenetnél megakadályozza, hogy az F_1 jelű tároló átváltson;
- az F_3 jelű tároló visszaállítást a T bemenetén keresztül a 9 dekódolása segítségével oldjuk meg.

A 4.10 táblázat az egyes flip-flopok logikai vezérlőjeleinek alakulását szemlélteti, amelynek segítségével nyomon követhetjük a számláló működését. A táblázatból megfigyelhető, hogy a 9 -es állapot felvétele után a számláló a következő órajelre a 0 -s állapotba kerül (tehát a számláló decimális típusú).



4.71. ábra. Szinkron decimális előreszámláló

Decimális érték	ENT T engedélyező	k				(k+1)				RCO				
		Q_3	Q_2	Q_1	Q_0	T_0	T_1	T_2	T_3		Q_3	Q_2	Q_1	Q_0
		2^3	2^2	2^1	2^0						2^3	2^2	2^1	2^0
0	1	0	0	0	0	1	0	0	0	0	0	0	1	0
1	1	0	0	0	1	1	1	0	0	0	0	1	0	0
2	1	0	0	1	0	1	0	0	0	0	0	1	1	0
3	1	0	0	1	1	1	1	1	0	0	1	0	0	0
4	1	0	1	0	0	1	0	0	0	0	1	0	1	0
5	1	0	1	0	1	1	1	0	0	0	1	1	0	0
6	1	0	1	1	0	1	0	0	0	0	1	1	1	0
7	1	0	1	1	1	1	1	1	1	1	0	0	0	0
8	1	1	0	0	0	1	0	0	0	1	0	0	1	0
9→0	1	1	0	0	1	1	0	0	1	0	0	0	0	1

4.10. táblázat. A flip-flopok logikai vezérlőjeleinek alakulása

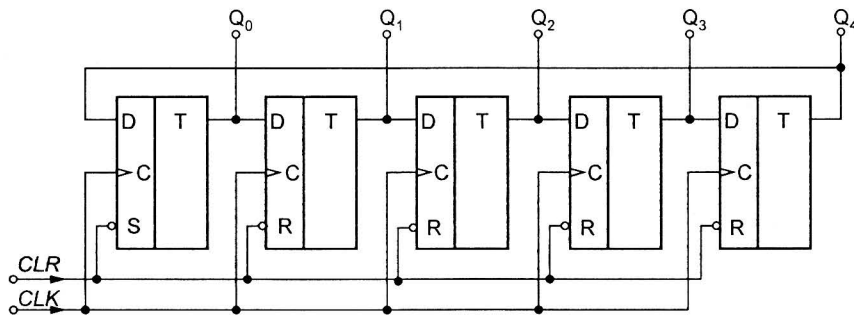
4.5.3. Gyűrűs számláló áramkörök

A gyűrűs számláló áramkörök felépítése és működése eltér az eddig ismertetett számlálókétól. A gyűrűs számláló tulajdonképpen visszacsatolt léptetőregiszterek, amelyekben az állapotok számát a visszacsatolás módja határozza meg. Mivel a léptetőregiszterek flip-flopjai közös órajelnek kapnak (vagyis egyszerre változtatnak állapotot), a gyűrűs számláló működési sebessége a szinkron számlálókkal egyenértékű.

A gyűrűs számlálókat a kódolás szempontjából csoportosíthatjuk, a következőképpen:

- „ N -ből 1” kódban működő számláló;
- Johnson-számláló;
- maximális hosszúságú számláló.

Az „ N -ből 1” kódban működő számláló információtartalma 1 bit kivételével azonos. A számláló állapotát az jelzi, hogy a többitől eltérő bit melyik flip-flopban van. A 4.72. ábra egy „5-ből 1” kódban működő gyűrűs számláló kapcsolását mutatja.



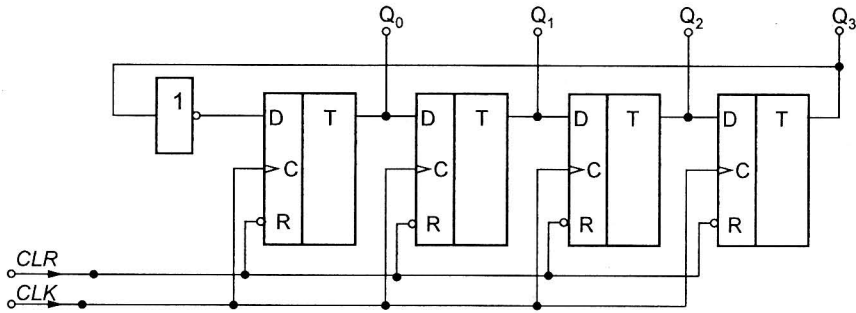
4.72. ábra. N -bites léptetőregiszter

CLR (clear) – törlés (logikai 0-ban aktív)

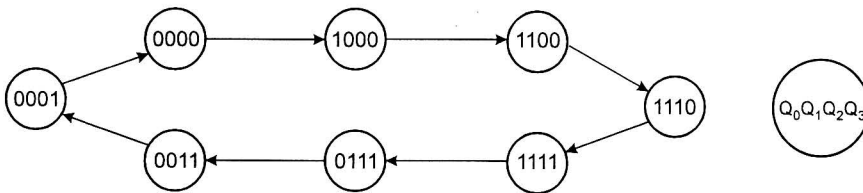
CLK (clock) – órajel

A CLR bemenetre adott logikai 0 érték, nullázza a számlálót (vagyis $Q_0 = 1$ lesz, a többi kimenet pedig logikai 0 állapotba kerül). Ezután minden órajelre a regiszter tartalma egy helyértéssel jobbra lép. Az így kapott számláló áramkör előnye, hogy kombinációs hálózatot nem tartalmaz és ennek következtében működése gyorsabb, mint a kapuáramkörökkel is tartalmazó szinkron számlálóké. Ugyanakkor előnynek tekinthető, hogy a számlálás eredményét nem BCD kódban kapjuk meg, hanem a számlálás eredményét a logikai 1-es szintű kimenet pozíciója közvetlenül jelzi. Hátránya az áramkörnek (ami miatt ritkán használják), hogy a megkülönböztethető állapotok száma ($modulusa$) azonos a flip-flopok számával, tehát nagy állapotszám esetén igen sok flip-flop kell a megvalósításhoz.

A Johnson-számlálókval gazdaságosabban alakíthatók ki nagy modulusú számláló áramkörök. A 4.73.a ábra egy 4-bites Johnson-számláló kapcsolását, a 4.73.b ábra a számláló állapotdiagramját mutatja.



a) kapcsolás



b) állapotdiagram

4.73. ábra. 4-bites Johnson-számláló*CLR (clear) – törlés**CLK (clock) – órajel*

Az áramkör számlálási ciklusa 8 állapotot tartalmaz. Az állapotsorozat jellegzetessége, hogy a számláló balról jobbra logikai 1-ekkel, majd 0 bitekkel töltődik fel. N számú flip-flop esetén az 1 és 0 értékű bitekkel való feltöltődés $N-N$ különböző állapotot eredményez.

Megállapítható, hogy N számú flip-flop (N -bites léptetőregiszter) esetén a különböző állapotok száma:

$$2 \cdot N.$$

A Johnson-számláló az „ N -ből 1” kódhoz képest tehát kétszer több állapotot biztosít. Hátránya viszont az áramkörnek, hogy kijelzés cáljaira dekóder áramkör alkalmazása szükséges.