

# Számlálók és frekvenciaosztók

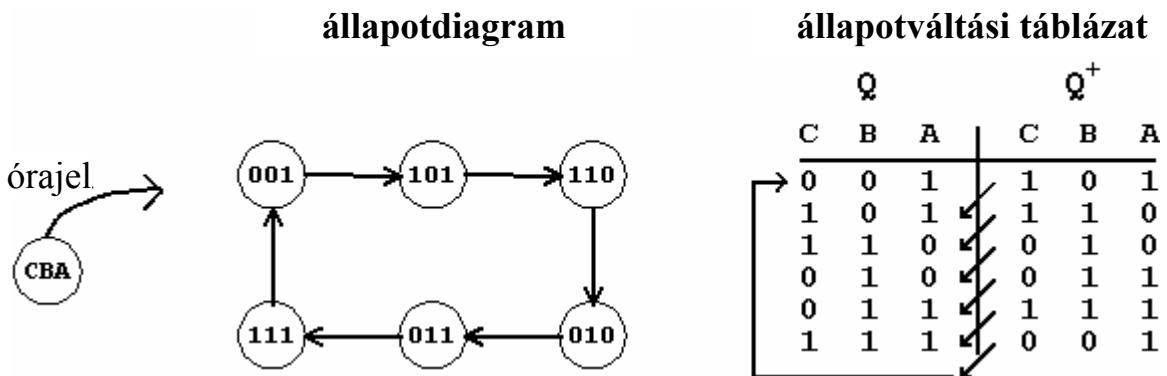
## Szinkron, aszinkron számlálók

A szekvenciális hálózatok egyik legfontosabb csoportja a számlálók. Hasonlóan az „1” és „0” jelölésekhez a számlálók kimenetei sem interpretálандók mindig számként, pl. a kimeneteikkel más logikai elemeket lehet vezérelni.

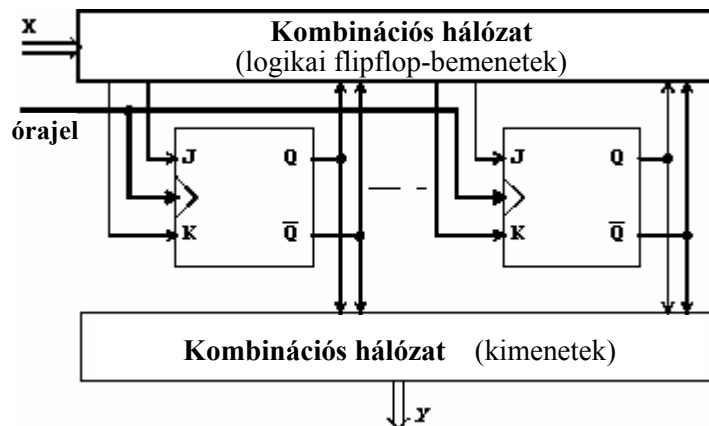
A számlálók lényegében egymást követő állapotokon mennek keresztül megadott sorrendben és a mindenkorli állapothoz tartozó bitmintát adják ki a kimeneteiken, melyek például számokat is jelenthetnek.

A számlálók működését is állapotdiagramokkal lehet leírni.

**Példa**-számláló, állapotváltás a hatásos órajel-ével:



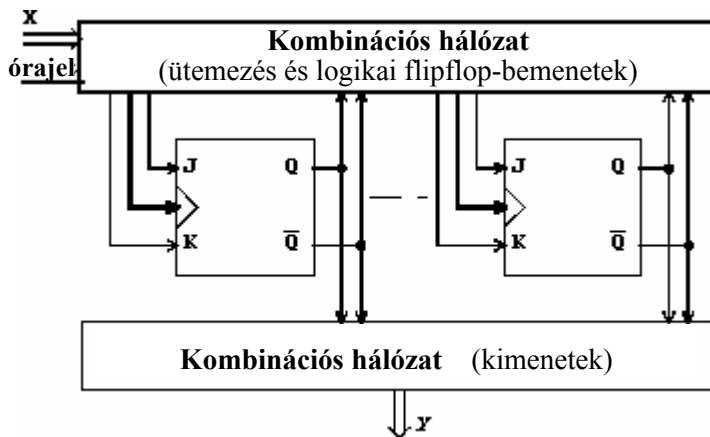
**Szinkron számlálóknál** az egyes állapot-bitek egyidőben, a közös órajel által szinkronizálva változnak. (A számlálók általában az órajel beérkező impulzusait számolják.) Minden hatásos órajel-él állapotváltást vált ki, és minden állapotváltásnál minden flipflop kap egy hatásos órajel-impulzust.



# Számlálók és frekvenciaosztók

## Szinkron, aszinkron számlálók

**Aszinkron üzemben** az egyes állapot-biteket tároló flipflop-ok ütemjele az adott állapotváltás idejére hatástalanítható, amennyiben az adott állapotváltásnál ennek az adott bitnek nem kell megváltozni.



A tényleges aszinkron viselkedés abból ered, hogy az egyes bit-értékek megtartását nem egyszerűen az órajel kiiktatásával oldjuk meg, hanem az egyes flipflopok órajelét a kombinációs hálózat képezi és a sorban következő flipflop ütemezéséhez az előző flipflop kimeneti jelét használja. Ezáltal az egymás után következő FF-ok ütemezése a soron végigterjed, időben eltolódik.

- Szinkron számlálók egyszerűbb felépítésűek, könnyebben kibővíthetőek. Kombinációs hálózatuk bonyolultabb, viszont nincsenek jelterjedési időből fakadó problémák.
- Az aszinkron számlálók kombinációs hálózata kevesebb kapuáramkörből áll, de mint minden aszinkron folyamatnál, itt is felléphetnek áttekinthetetlen jelterjedési idő-problémák.

---

# Számlálók és frekvenciaosztók

## Szinkron, aszinkron számlálók

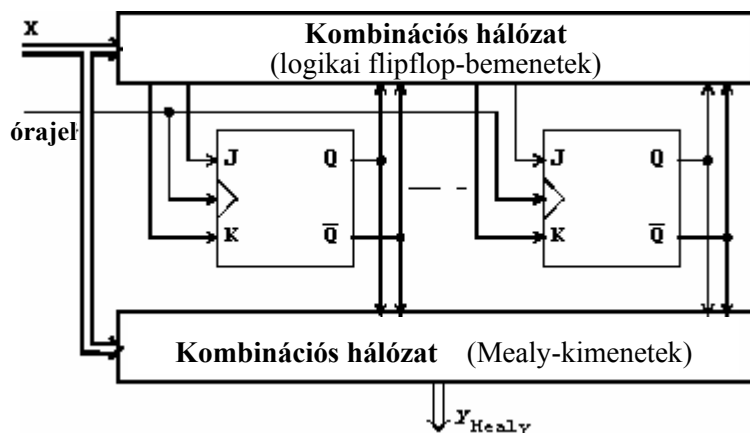
---

A **kimeneti kombinációs hálózat** opcionális: az állapotkódot alakítja át a kívánt kimeneti kóddá. A kettő különválasztásának előnye, hogy tervezéskor egyszerűbb állapotkódokat lehet betervezni (az állapotváltáskor fellépő bit-változások száma alacsony legyen).

A számlálók **bemeneteit** két fő csoportra bonthatjuk:

- **órajel**, (számlálandó impulzusok)
- **X**, vezérlő jelek (pl. felfele- ill. lefele-számlálás, speciális tárolási. ill. beírási funkciók, lépéstávolság definiálása, stb.)

Léteznek számlálók, ahol a kimenet nem csak az aktuális állapottól függ, hanem az X vezérlő jelektől is. Az állapotkódot ekkor a Mealy-automatákhoz hasonlóan közvetlenül össze kell kötni a bemenetekkel a kimeneti kombinációs hálózatban. Ez a kimeneti KH csak a Moore-elven felépített számlálónál hagyható el (kimenetkód = állapotkód).



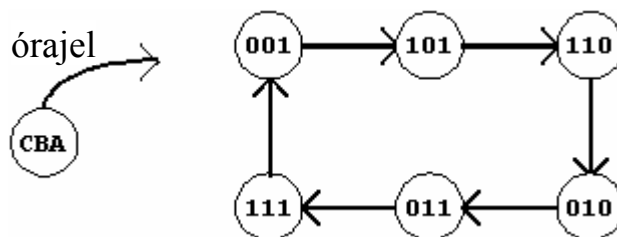
Mealy-kimenettel rendelkező számlálóknál az X bemeneten jelentkező esetleges zavarimpulzusok közvetlenül hatnak az Y kimenetre, ellentétben a Moore-féle számlálókkal, ahol a flipflopok állapot-tartása (pufferhatás) miatt a zavarimpulzusok nem terjednek tovább.

# Számlálók és frekvenciaosztók

## Szinkron számlálók tervezése

### Példa: Modulo-6-számláló tervezése

Adott az állapotdiagram:



A felső határfrekvenciának (számlálás gyorsaságának) növelése érdekében ne legyen kimeneti kombinációs hálózat : kimeneti kód megegyezik az állapotkóddal.

### Fejlesztési fázis (két lépésben)

(„top-down”-módszer: általános leírásból kiindulva eljutni a konkrét kapcsolásig)

#### 1. lépés: a flipflop-darabszám és az állapotkódok meghatározása

Adott m számlálókapacitás („modulo-m”) esetében bináris kódot használva legalább n db flipflop szükséges:

$$m \leq 2^n$$

A számlálásnál használt kódok többsége nem ennyire „gazdaságos” és ennél több flipflop-ot igényel (pl. n-ből 1 kód, egylépéses kódok stb.)

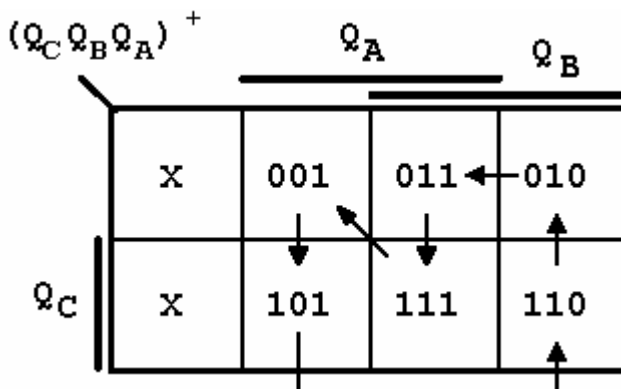
A példában 3 flipflop-ra van szükség:  $2^3 = 8$

‘000’ és ‘100’ pszeudotriadok

#### 2. lépés: a flipflop-típusok kiválasztása és a KH megtervezése

A számlálók többnyire élvezérelt D-flipflop-ból, vagy a több funkcióval rendelkező élvezérelt JK-flipflop-ból épülnek fel .

Először flipflop-típustól függetlenül állapotváltási-diagramban ábrázoljuk az egyes állapotok sorrendiségét, kiindulva az állapotdiagramból. Ezután a kiválasztott flipflop-ra jellemző KV-diagram összeállítása következik:



# Számlálók és frekvenciaosztók

## Szinkron számlálók tervezése

### Példa-számláló (modulo-6) megvalósítása D-flipfloppal

A  $D_i$ -bemenetekre kell felírni a KV-diagramokat, melyből az összes FF-bemenetre a kapcsolás kiolvasható.

A D-flipflop-ra érvényes:  $Q_i^+ = D_i$

$D_A$	$Q_A$			$Q_B$
	X	1	1	1
$Q_C$	X	0	1	0

$D_B$	$Q_A$			$Q_B$
	X	0	1	1
$Q_C$	X	1	0	1

$D_C$	$Q_A$			$Q_B$
	X	1	1	0
$Q_C$	X	1	0	0

Az adott állapotbitet képviselő flipflop bemenetének KV-diagramját úgy kell kitölteni, hogy minden egyes állapot-cellába az az érték kerüljön, amelyiket az adott FF-(állapot)bit a következő órajel-élel felvesz.

A  $D_i$ -bemenetek logikai függvénye a fentiek alapján, a lehetséges összevonásokkal:

$$D_A = \bar{Q}_C + Q_A \cdot Q_B$$

$$D_B = \bar{Q}_A + Q_B \oplus Q_C$$

$$D_C = \bar{Q}_B + Q_A \cdot \bar{Q}_C$$

### Példa-számláló megvalósítása JK-flipfloppal

Az egyes JK-flipflop-ok állapotváltási-diagramját célszerű a JK-flipfloptípushoz igazítva úgy felírni, hogy az állapotokat képviselő cellákba az adott FF-bit soron következő változását is feltüntetjük:

$Q_A^+$	$Q_A$			$Q_B$
	X	1→1	1→1	0→1
$Q_C$	X	1→0	1→1	0→0

$Q_B^+$	$Q_A$			$Q_B$
	X	0→0	1→1	1→1
$Q_C$	X	0→1	1→0	1→1

$Q_C^+$	$Q_A$			$Q_B$
	X	0→1	0→1	0→0
$Q_C$	X	1→1	1→0	1→0

# Számlálók és frekvenciaosztók

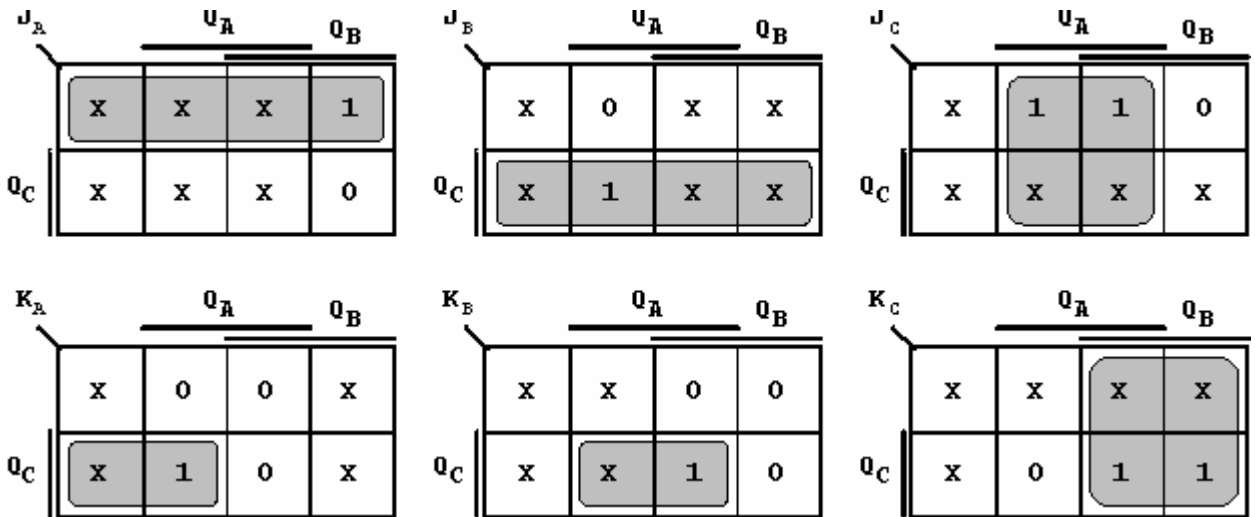
## Szinkron számlálók tervezése

**Példa**-számláló megvalósítása JK-flipfloppal (folytatás)

JK-flipflop állapotváltási táblázata:

$Q \rightarrow Q^+$	J	K	Funkció
$0 \rightarrow 0$	0	x	not jump
$0 \rightarrow 1$	1	x	jump
$1 \rightarrow 0$	x	1	kill
$1 \rightarrow 1$	x	0	not kill

KV-diagramok:



A KV-diagramokból levezethető  $J_i$  és  $K_i$  logikai függvények:

$$J_A = \bar{Q}_C$$

$$K_A = Q_C \cdot \bar{Q}_B \quad (\bar{K}_A = \bar{Q}_C + Q_B)$$

A J-K-bemenetek vezérlését a JK-flipflop logikai egyenletéből is levezethetjük:

Mivel  $Q^+ = J_i \cdot \bar{Q}_i + \bar{K}_i \cdot Q_i \rightarrow Q_A^+ = J_A \cdot \bar{Q}_A + \bar{K}_A \cdot Q_A$   
 Az állapotváltási-diagramból következik:  $Q_A^+ = \bar{Q}_A \cdot \bar{Q}_C + Q_A \cdot (Q_B + \bar{Q}_C)$

A két egyenlet összehasonlításával szintén a fenti eredményt kapjuk.

$Q_B$  és  $Q_C$  flipflopok J-K-bemeneteire a függvényeket szintén a fenti eljárásokkal kaphatjuk meg.

---

# Számlálók és frekvenciaosztók

## Aszinkron számlálók tervezése

---

Az aszinkron számlálók tervezésénél a szinkron 1. és 2. lépés közzé egy harmadik lépést kell beiktatni, mely a flipflop-ok ütemezését szabályozza.

### 1. lépés: a flipflop-darabszám és az állapotkódok meghatározása

Ez a fejlesztési fázis megegyezik a szinkron számlálókéval.

### 2. lépés: Ütemezés

Minden egyes számlálóimpulzusra meg kell határozni, hogy melyik FF változtasson állapotot. Csak ezek a FF-ok kapnak órajel-impulzust. Azon FF-oknál, melyek állapota az adott számlálóimpulzusra nem változik, - mivel nem kapnak órajel-impulzust - lényegtelen a logikai bemenetek szintje: „don't care”- egyszerűsítési lehetőség a KH-ban.

### 3. lépés: A kombinációs hálózat (KH) tervezése

A KH-nak a logikai bemeneteken túlmenően az órajel-bemenetek jelét is elő kell állítania.

Az aszinkron működési mód miatt egy FF-váltás után közvetlenül a KH-ban egyidőben léteznek régi és új bemenő jelek. Az új jeleknek megfelelő KH-kimenetek és az ez alapján beálló számláló-kód csak bonyolult jelterjedési folyamatok lecsengésekor, több köztes-állapot után stabilizálódnak.

#### Megjegyzések:

- Tervezéskor figyelmen kívül kell hagyni a jelterjedési időket. Egyik állapotból a másikba történő váltás tervezésekor csak a régi állapot értékeit vegyük alapul.
- Mindegyik számláló-állás az egyes flipflopok aszinkron (egymásutáni) átbillenésének következtében áll be („*ripple-changing*”).

# Számlálók és frekvenciaosztók

## Aszinkron számlálók tervezése

### Példa: Modulo-16-számláló tervezése

Adott az állapotváltási táblázat :

Meg kell vizsgálni, létezik-e olyan FF-kimenet-váltás, melyet egyazon állapotváltásnál egy vagy több másik flipflop ütemjeleként közvetlenül fel lehetne használni. Ezek szerint:

FF órajel-bemenet	órajel
$C_A$	külső órajel
$C_B$	$Q_A$
$C_C$	$Q_B$
$C_D$	$Q_C$

órajel	Q				Q <sup>+</sup>			
	D	C	B	A	D	C	B	A
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	1	0	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	1	0
6	0	1	1	0	0	1	1	1
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	1
9	1	0	0	1	1	0	1	0
10	1	0	1	0	1	0	1	1
11	1	0	1	1	1	1	0	0
12	1	1	0	0	1	1	0	1
13	1	1	0	1	1	1	1	0
14	1	1	1	0	1	1	1	1
15	1	1	1	1	0	0	0	0

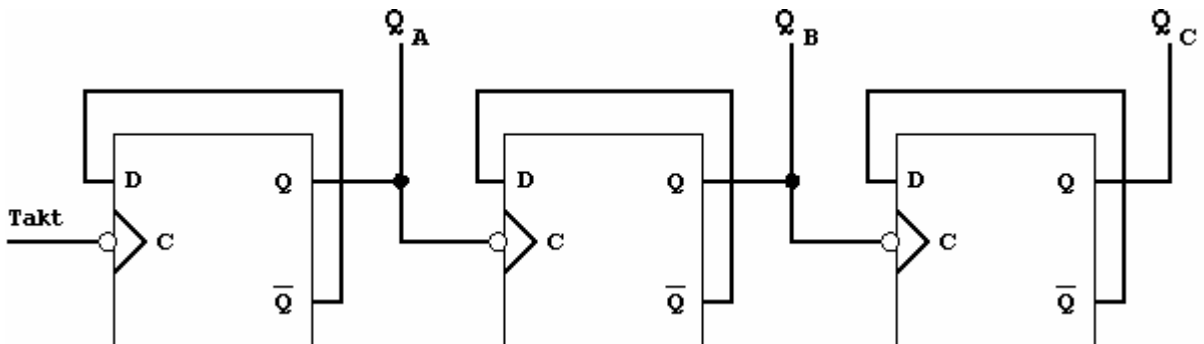
**FF<sub>A</sub>**: ez a FF minden órajellel változik, tehát a külső órajellel kell vezérelni

**FF<sub>B</sub>**: B flipflop a  $Q_A:1 \rightarrow 0$  átmenetkor változik, tehát az ütemezést  $Q_A$  végzi

**FF<sub>C</sub>**: C flipflop a  $Q_B:1 \rightarrow 0$  átmenetkor változik, tehát az ütemezést  $Q_B$  végzi

**FF<sub>D</sub>**: D flipflop a  $Q_C:1 \rightarrow 0$  átmenetkor változik, tehát az ütemezést  $Q_C$  végzi

A FF-ok ebben a példában minden órajelre csak a „váltás” (*toggle*) funkciót végzik. A számlálót ezért vagy T-flipflopok vagy megfelelően visszacsatolt D-flipflopok sorbakötésével valósíthatjuk meg (lefutó órajel aktív).





# Számlálók és frekvenciaosztók

## Aszinkron számlálók tervezése

### Példa: BCD-számláló tervezése („modulo-10”)

Adott az állapotváltási táblázat :

A tervezéshez két lehetőség kínálkozik:

Megvalósítás

- a már ismert modulo-16-számláló módosításával
- az aszinkron számlálók általános tervezési módszerével

órajel	Q				Q <sup>+</sup>			
	D	C	B	A	D	C	B	A
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	1	0	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	1	0
6	0	1	1	0	0	1	1	1
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	1
9	1	0	0	1	0	0	0	0

### BCD-számláló megvalósítása modulo-16-számláló módosításával

(A két számláló állapotváltási táblázatának hasonlósága miatt) ki lehet indulni a modulo-16-számlálóból. Módosításra (az ütemezés módosítására) a BCD-számláló átfordulási határának elérésekor van szükség.

**Szükséges változtatások:**

**B-flipflop :**

A 0 → 1 átmenet elkerülésére egyik lehetőség a D-bemenet „0”-ra állítása, viszont aszinkron számlálókra a másik megoldás, az ütemjel kiiktatása az inkább jellemző:  $C_B = Q_A$  addig legyen érvényes, amíg  $Q_D = 0$

Megoldás:  $C_B = Q_A \cdot \overline{Q_D}$

**D-flipflop :**

Az eddigi ütemjelet ( $Q_C$ -t) ki kell egészíteni arra az esetre is, amikor  $Q_A = Q_D = 1$ .

Megoldás:  $C_D = Q_C + Q_A \cdot Q_D$

órajel	Q				Q <sup>+</sup>			
	D	C	B	A	D	C	B	A
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
·		·				·		
·		·				·		
·		·				·		
8	1	0	0	0	1	0	0	1
9	1	0	0	1	0	0	0	0
9	1	0	0	1	1	0	1	0

Mod-10-számláló

Mod-16-számláló

↑            ↑  
FFD        FFb

# Számlálók és frekvenciaosztók

## Aszinkron számlálók tervezése

### BCD-számláló megvalósítása aszinkron számlálók általános tervezési módszerével

Állapot-átváltási diagram a szükséges négy FF-kimenetre:

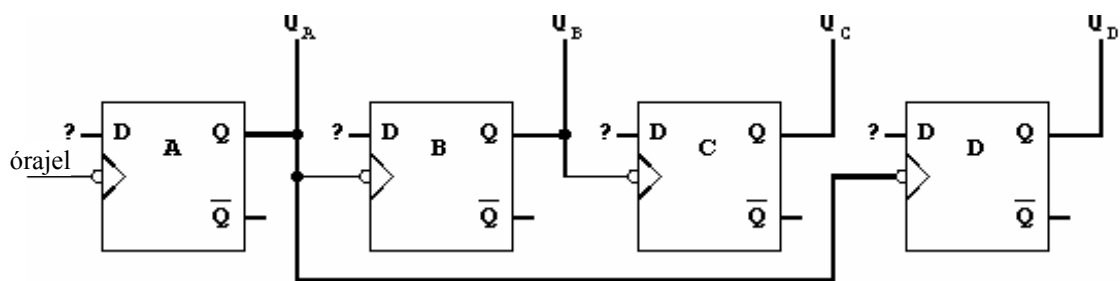
Az állapot-váltási táblázatból levezethető ütemezés:

Flipflop	órajel
A	mindegyik (0-9)
B	1,3,5,7
C	3,7
D	7,9

$(Q_D Q_C Q_B Q_A)^+$

	$Q_A$		$Q_B$	
	0	1	3	2
	4	5	7	6
$Q_C$	X	X	X	X
$Q_D$	8	9	X	X

A választott alapkapsolás:  
(többlet-ütemezést bekalkulálva)



Fenti alapkapsolás ütemjel-táblázata,  
a nem kívánt ütemjelek zárójelben:

Flipflop	ütemjel
A	mindegyik
B	1,3,5,7,(9)
C	3,7
D	(1),(3),(5),7,9

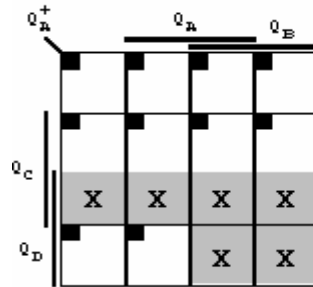
# Számlálók és frekvenciaosztók

## Aszinkron számlálók tervezése

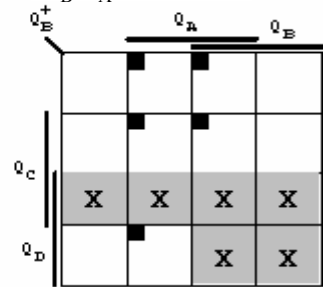
### BCD-számláló megvalósítása aszinkron számlálók általános tervezési módszerével (folytatás)

Az egyes FF-kimenetek KV-diagramja, ■ jellel jelölve azokat az állapotokat, amelyek fennállásakor a következő órajellel az adott FF-kimenet átbillen.

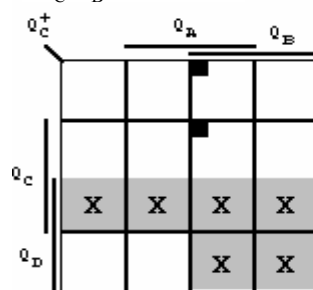
FF<sub>A</sub> az eredeti órajellel ütemezve



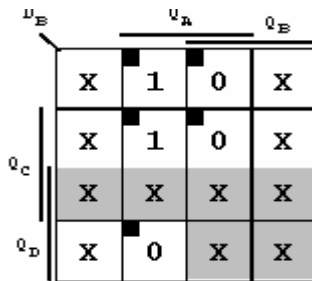
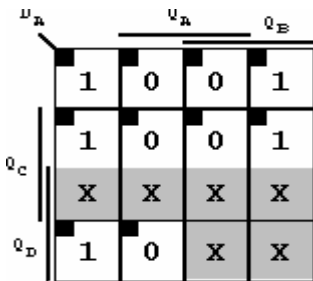
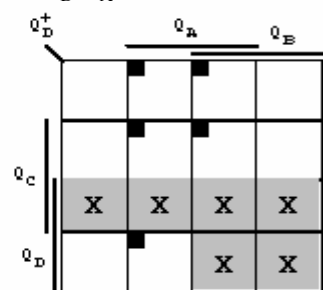
FF<sub>B</sub>  $Q_A$ -val ütemezve



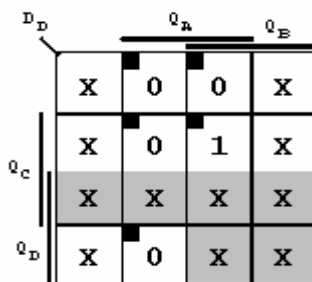
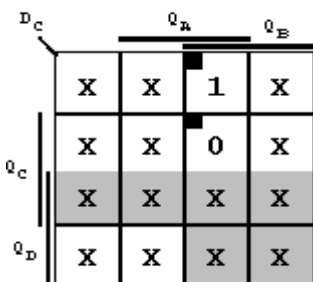
FF<sub>C</sub>  $Q_B$ -vel ütemezve



FF<sub>D</sub>  $Q_A$ -val ütemezve



A következő lépésben a konkrét FF-típus vezérlési sajátosságait figyelembe véve adódik a tényleges KV--diagram (itt. D-flipflop)

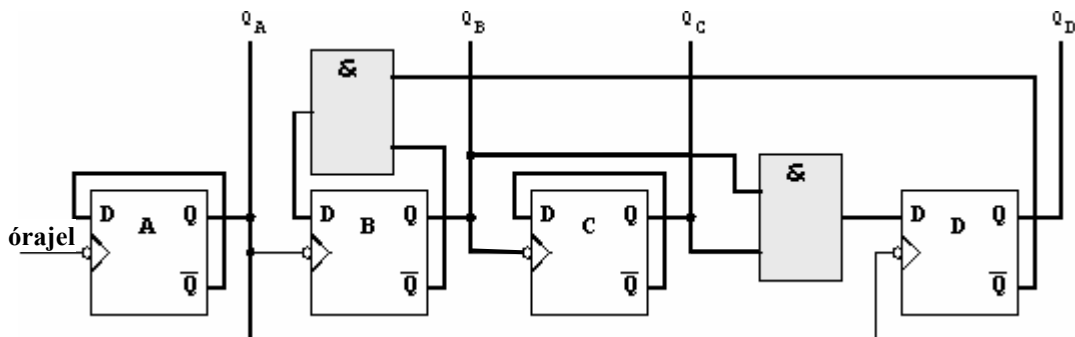


$$D_A = \overline{Q_A}$$

$$D_B = \overline{Q_B} \cdot \overline{Q_D}$$

$$D_C = \overline{Q_C}$$

$$D_D = Q_B \cdot Q_C$$



# Számlálók és frekvenciaosztók

## Aszinkron számlálók tervezése

### Aszinkron számlálók tervezési fázisai: (összefoglalás)

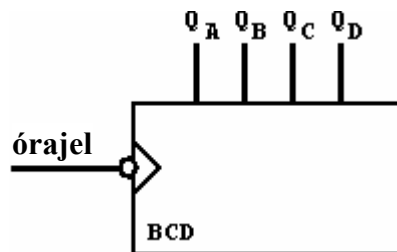
- A megvalósítani kívánt számláló leírása az állapotváltási táblázattal és az állapotváltási diagrammal
- A szükséges flipflop-darabszám meghatározása
- A szükséges ütemezések meghatározása
- A rendelkezésre álló órajel-források meghatározása
- Flipflop-független KV-diagram meghatározása a ténylegesen fellépő órajelek figyelembevételével
- A FF-típus kiválasztása és a FF-specifikus KV-diagramok felállítása
- A KV-diagramokból az optimalizált logikai függvények kiolvasása
- A kapcsolás megvalósítása

### Többszintű számlálók

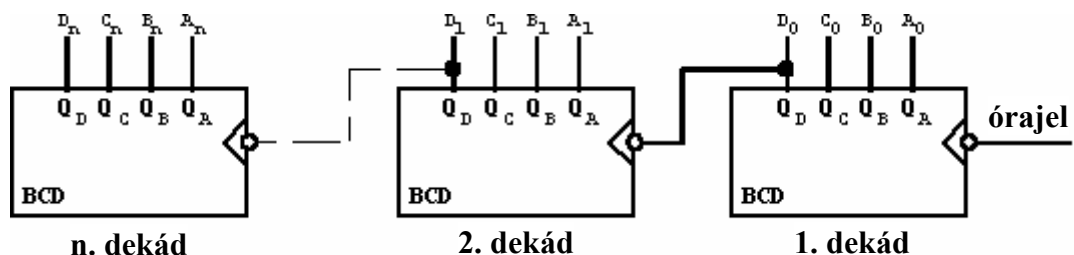
Az egyes fokozatok sorba kapcsolásával jönnek létre.

### Példa: aszinkron BCD-számláló megvalósítása

Egy számlálófokozat kapcsolási jele (nem szabvány):  
számlálókapacitás: 0 - 9



Pl. egy három-helyiértékes BCD-számláló (három dekád)  
számlálókapacitása: '000' - '999'



# Számlálók és frekvenciaosztók

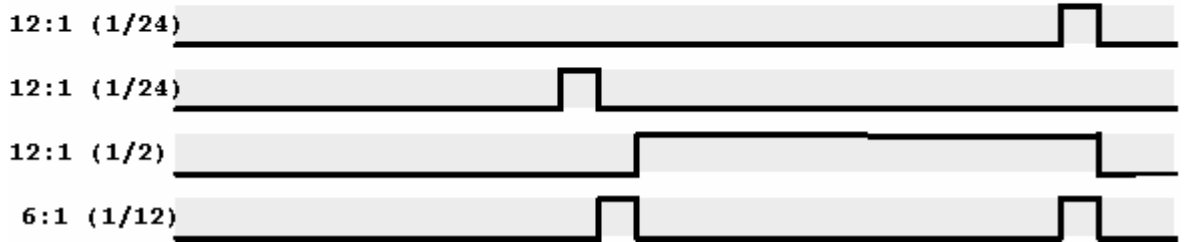
## Frekvenciaosztók

A frekvenciaosztók is impulzusokat számolnak mint a számlálók, azzal a különbséggel, hogy nem adják ki a kimenetükre minden egyes belső számláló-állapotot. Bizonyos számú beérkezett impulzus után adnak ki csupán egy kimeneti impulzust.

bemeneti impulzus-sorozat:



kimeneti impulzus-sorozat:



osztási arány (kitöltési tényező)

**Osztási arány:** a bemeneti impulzusok és a kimeneti impulzusok darabszámának aránya

$$T_f = \frac{f_{be}}{f_{ki}}$$

**Kitöltési tényező (duty cycle):** az impulzus ideje és a periódusidő aránya

$$T_p = \frac{t_{high}}{t_{high} + t_{low}}$$

Gyakorlatilag frekvenciaosztókat önálló frekvenciaosztók sorbakötésével is megvalósíthatunk. Univerzálisan használhatók például a prím-osztók: 2:1, 3:1, 5:1, 7:1, 11:1, 13:1, stb. (Például 20:1 osztó megvalósítható 2 db 2:1, és 1 db 5:1-osztó sorbakötésével.) Több osztó sorbakötésének hátránya az alacsony felső határfrekvencia.

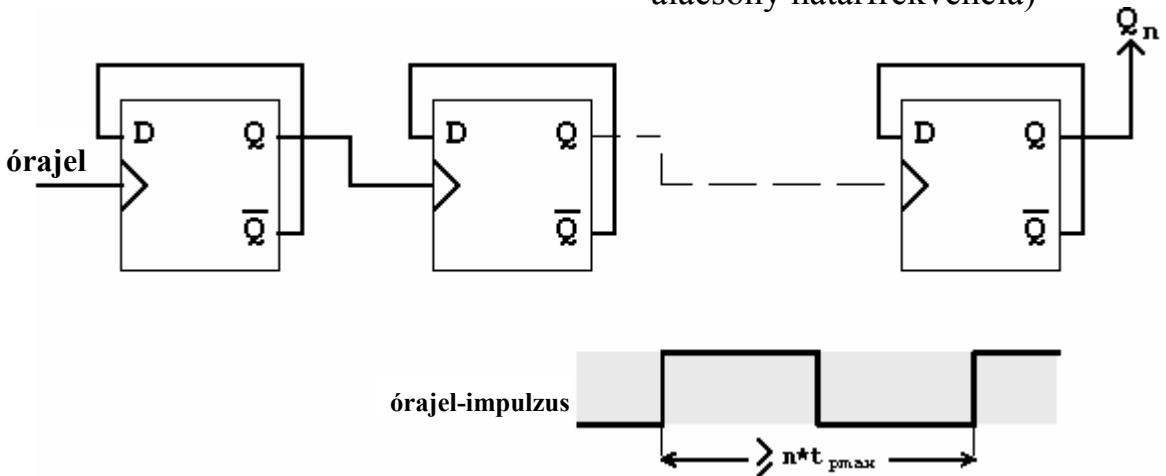
A frekvenciaosztók két alapvető csoportra bonthatók, melyek alapfelépítésükben is különböznek:

- osztási arány „ $2^n : 1$ ”
- osztási arány „ $m : 1$ ”, ahol  $m < 2^n$

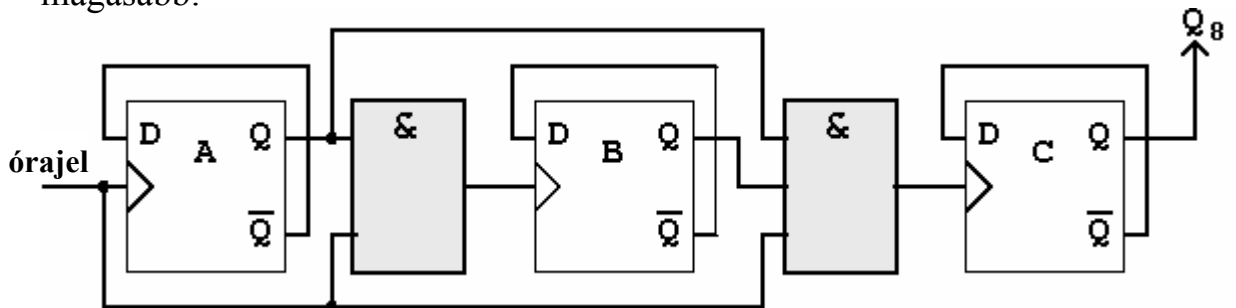
# Számlálók és frekvenciaosztók

## Frekvenciaosztók, osztási arány $2^n:1$ , aszinkron

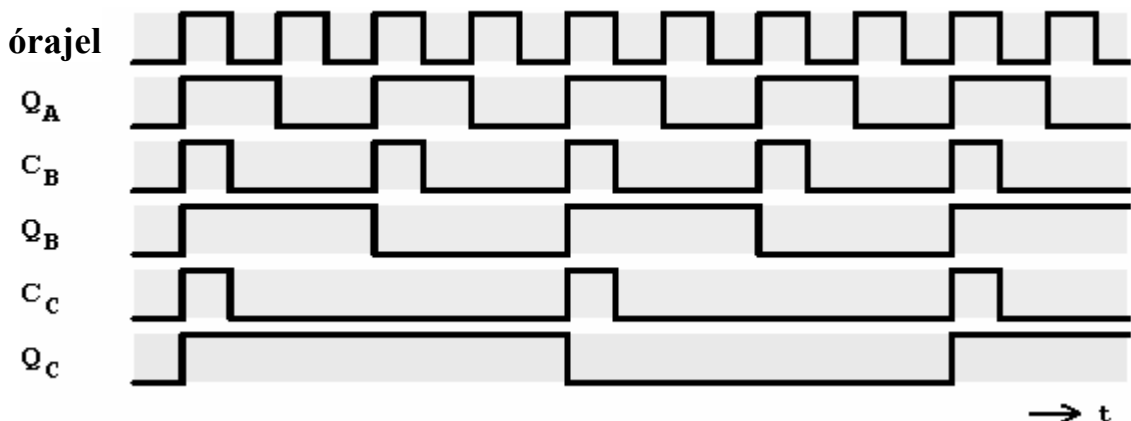
T-flipflopok sorbakötésével megvalósított aszinkron frekvenciaosztó  
2:1, 4:1, 8:1, ...,  $2^n:1$  osztási arányokra: (egyszerű felépítés, viszont  
alacsony határfrekvencia)



Kvázi-aszinkron felépítésű frekvenciaosztóknál a felső frekvenciahatár  
magasabb:



A fenti 8:1 frekvenciaosztó impulzusdiagramja:

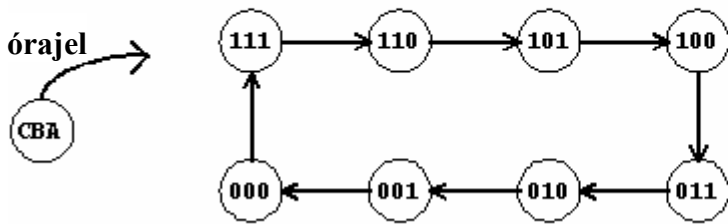


# Számlálók és frekvenciaosztók

## Frekvenciaosztók, osztási arány $2^n:1$ , szinkron

### $2^3:1$ szinkron frekvenciaosztó tervezése

állapotdiagram



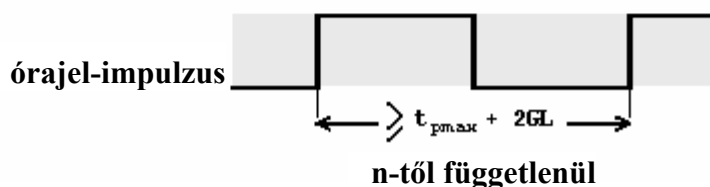
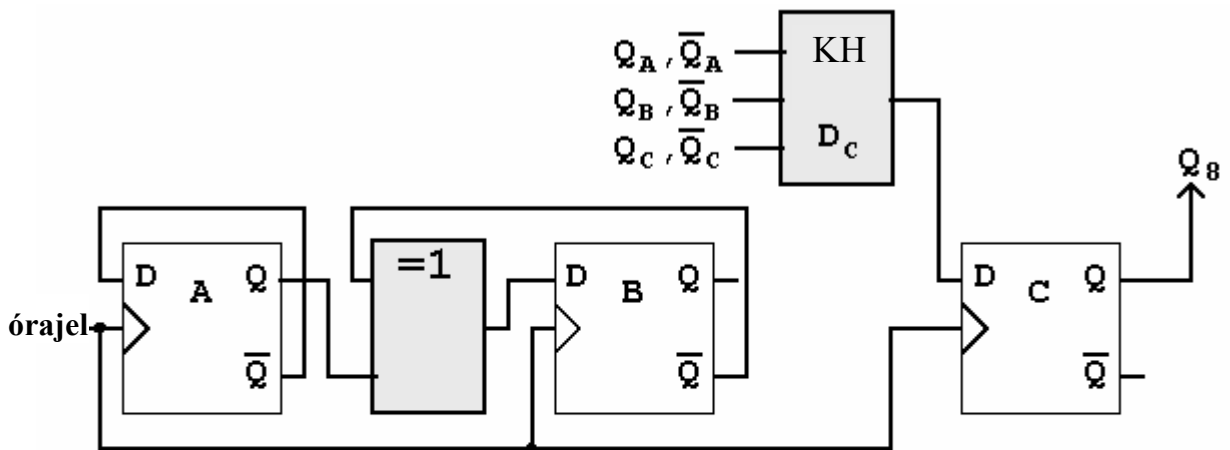
állapotváltási táblázat

	Q			Q <sup>+</sup>		
	C	B	A	C	B	A
1	1	1	1	1	1	0
1	1	0	1	1	0	1
1	0	1	1	1	0	0
1	0	0	1	0	1	1
0	1	1	1	0	1	0
0	1	0	1	0	0	1
0	0	1	1	0	0	0
0	0	0	1	1	1	1

KV-diagram B és C flipflopokra:  
(az A flipflop „toggle” üzemmódban működik)

D <sub>B</sub>	u <sub>A</sub>		Q <sub>B</sub>	
	1	0	1	0
Q <sub>C</sub>	1	0	1	0

D <sub>C</sub>	u <sub>A</sub>		Q <sub>B</sub>	
	1	0	0	0
Q <sub>C</sub>	0	1	1	1



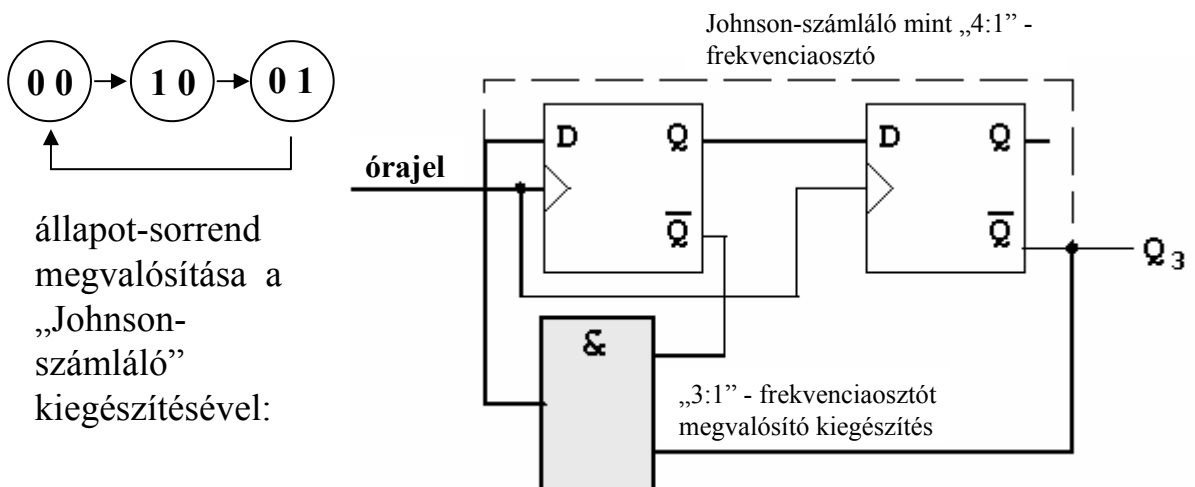
# Számlálók és frekvenciaosztók

## Frekvenciaosztók, osztási arány $m:1$ , aszinkron

„ $m:1$ ” frekvenciaosztóknál  $n$  db floplop-ra van szükség, ahol  $m < 2^n$

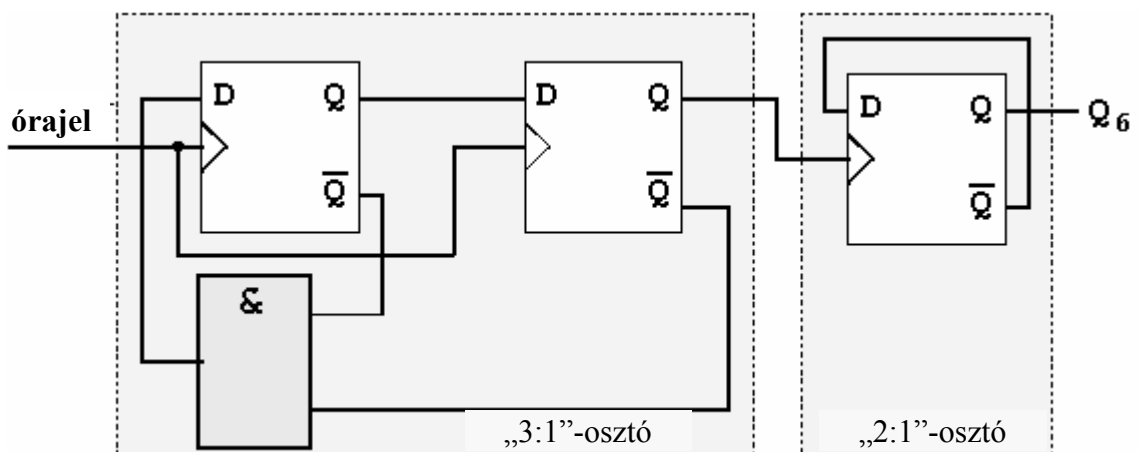
A tervezés lényege, hogy a  $2^n$  lehetséges állapotból  $m$  db-ot válasszunk ki, melyekkel a kívánt frekvenciaosztás optimálisan megvalósítható.

A 2-bites „Johnson”-számlálóból közvetlenül felépíthető „ $3:1$ ”-osztó kiindulási alaptípusként szolgál az egyik - (általánosan elterjedt) - „ $m:1$ ” szinkron frekvenciaosztó-típus tervezéséhez:



A Johnson számláló kiegészítése az AND-kapuvál gondoskodik arról, hogy csak a ‘00’-állapotban íródjon be sorosan egy ‘1’-es az első FF-ba, ezután kétszer léptetés következik.

6:1 számláló egy egyszerű „Toggle”-fokozat sorbakötésével:

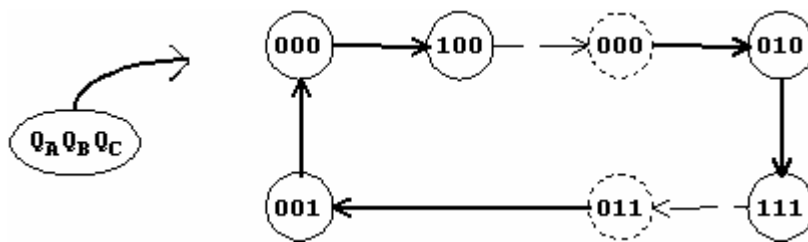
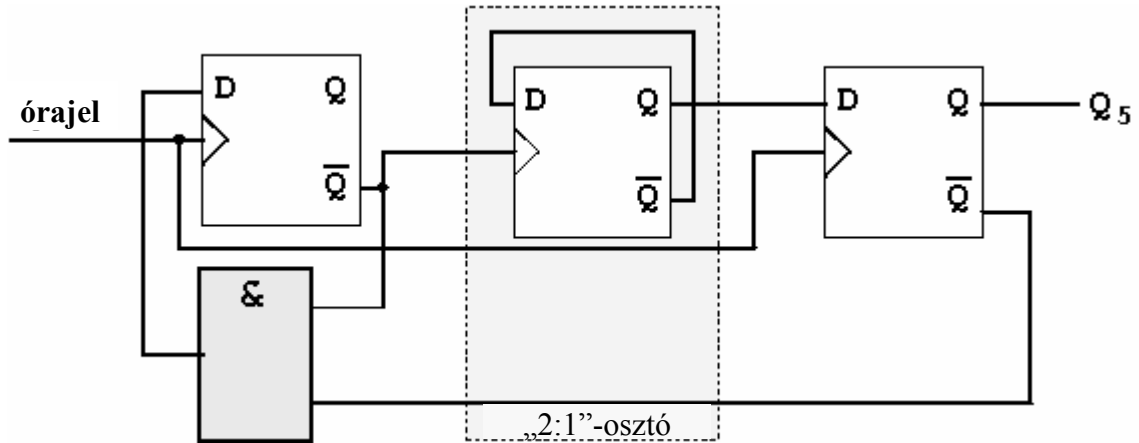




# Számlálók és frekvenciaosztók

## Frekvenciaosztók, osztási arány m:1, aszinkron

Ha a „Toggle”-flipflopot a „3:1”-osztó belsejébe helyezük el, egy „5:1”-osztót kapunk.



A köztes állapotok a „2:1”-osztó vezérlésének aszinkron jellege miatt,

$$\overline{Q}_A : 0 \rightarrow 1$$

átmenetkor jönnek létre.

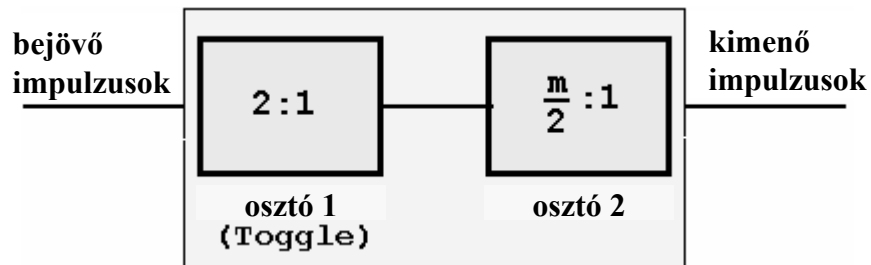
Ha az összes flipflop kimenetei ki vannak vezetve, az „m:1” frekvenciaosztó **m állapottal rendelkező számlálóként** is használható, amennyiben az előadódó kód megfelel.

# Számlálók és frekvenciaosztók

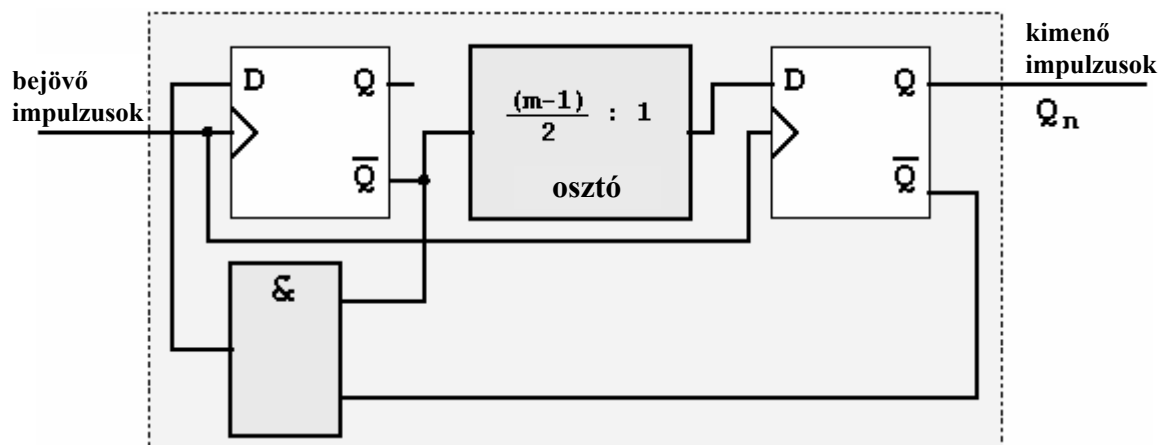
## Frekvenciaosztók, osztási arány $m:1$ , aszinkron

Összefoglalás: „ $m:1$ ” aszinkron frekvenciaosztók tervezési szabályai

- $m$  páros



- $m$  páratlan



**Példa:** "9:1"-frekvenciaosztó

