

# Szekvenciális hálózatok

## Állapotdiagram

### A kombinatorikus hálózatokra jellemző:

- A kimeneti paramétereket kizárólag a mindenkori bemeneti paraméterek határozzák meg, a hálózat jellegének, felépítésének megfelelően
- Nem rendelkeznek belső tároló elemekkel
- A kapuáramkörök nincsenek visszacsatolva.
- A hálózat bemenetén csak akkor változhat meg a jelszint, ha a megelőző összes jelváltozási folyamat a hálózaton belül lecsillapodott.

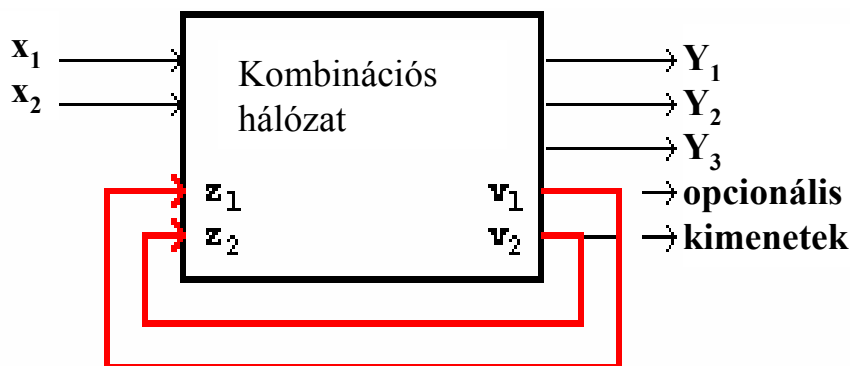
### A szekvenciális

#### kapcsolások

visszacsatolással rendelkeznek:

$$Y_i = f_{Y_i}(x_1, x_2, z_1, z_2)$$

$$V_i = f_{V_i}(x_1, x_2, z_1, z_2)$$



Állapotdiagram:

Bemenetek				Kimenetek				
szabad		visszacsatolt		belső		külső		
x <sub>1</sub>	x <sub>2</sub>	z <sub>1</sub>	z <sub>2</sub>	v <sub>1</sub>	v <sub>2</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>
0	0	0	0	1	1	1	1	1
0	0	0*	1*	0*	1*	1	0	0
0	0	1	0	1	1	0	1	1
:		:		:		:		
1	1	1*	0*	1*	0*	1	1	0
1	1	1	1	0	0	1	1	1

*\* stabil állapot*

A külső be- és kimenetek más hálózatokkal váltanak adatokat.

A visszacsatolt bemenetek jelét a hálózat maga állítja elő a belső kimeneten, (visszacsatolás) melyek jeleit más hálózatoknak is át lehet adni.

# Szekvenciális hálózatok

## Állapotdiagram

### Visszacsatolt hálózatok szekvenciális (sorrendi) működése:

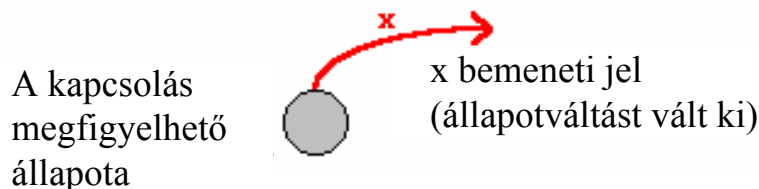
Stabil állapotban egy bemeneti jel -  $x_i$  - értéke megváltozik. Ha ennek következtében megváltozik valamelyik  $v_i$  visszacsatoló kimenet (bizonyos késleltetési idő után), akkor természetesen a  $z_i$  bemenet is megváltozik.

A további folyamatot illetően két eset lehetséges:

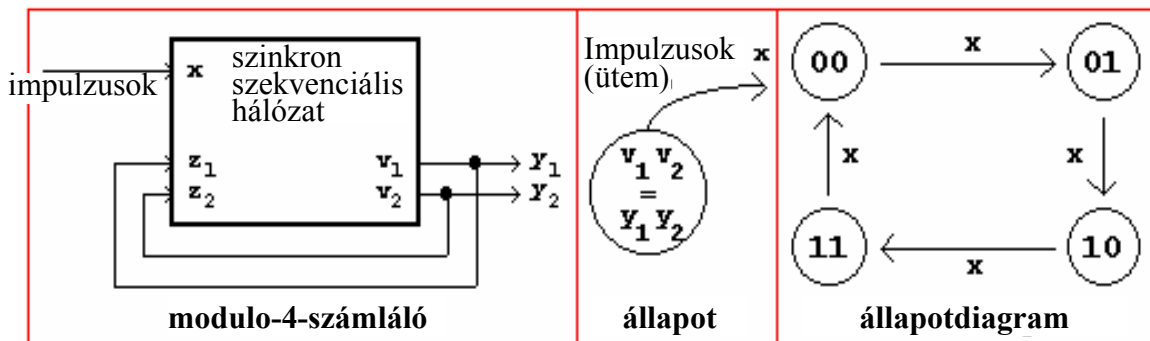
1.  $z_i$  új értéke következtében egyik  $v_i$  visszacsatoló kimenet sem változik meg. A kapcsolás stabil állapotba került.
2.  $z_i$  új értéke következtében újabb változás következik be az egyik  $v_i$  visszacsatoló kimeneten; a folyamat ismétlődik.

Általános esetben a jelek többször, egymás után áthaladnak a bemenettől a kimenetig, amit a pillanatnyi kapcsolás-állapotok sorozatával lehet jellemezni. Az állapotváltások sora addig tart, amíg a kapcsolás egy stabil állapotba nem kerül. A pillanatnyi állapot logikusan következik az előző állapotokból és a pillanatnyi külső bemenetekből.

### Szekvenciális hálózatok leírása állapotdiagrammal:



#### Példa: modulo-4-számláló



(A visszacsatolás és a kombinációs hálózaton belüli jel-áthaladási útvonalak különböző jelterjedési ideje miatt a hálózatban adott bemeneti jelek mellett bonyolult jelterjedési folyamatok játszódhatnak le → átmeneti zavarimpulzusok.)

# Szekvenciális hálózatok

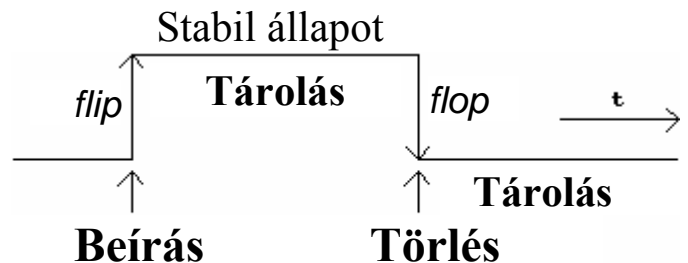
## Alap-Flipflop

A Flipflop (FF) áramkörök a szekvenciális hálózatok legegyszerűbb csoportját képezik.

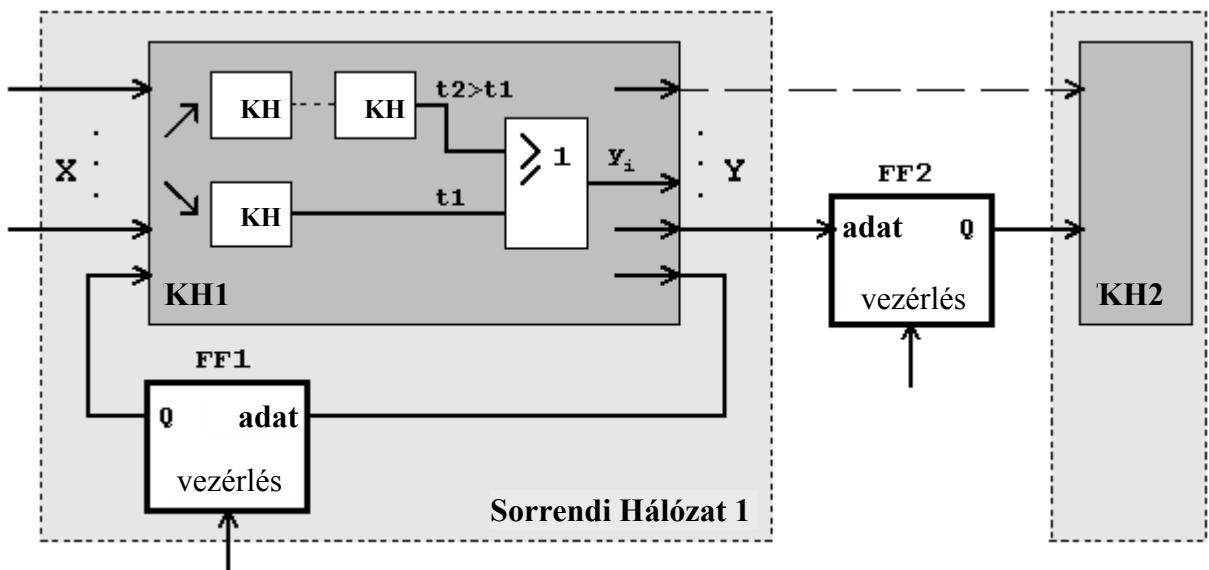
Belső tárolóval rendelkeznek, ami miatt a működésüket a pillanatnyi bemeneti jelszinteken túl a belső tároló aktuális állapota (azaz a FF „előélete”) is meghatározza.

A legegyszerűbb FF-kapcsolás három funkcióra alkalmas:

- **Beírás:** FF-kimenet „1”-re váltása
- **Törlés:** FF-kimenet „0”-ra váltása
- **Tárolás:** információ megőrzése



Jellemző flipflop alkalmazás:



FF1 és FF2 „kapcsolóként” működve biztosítják, hogy a kombinációs hálózatok bemenetén lévő jel ne változzon addig, amíg a KH-n belül az összes jelváltozási folyamat le nem csengett:

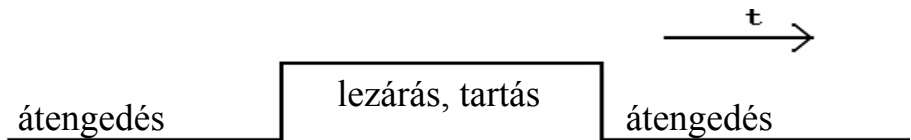
- Az „adat - Q” összeköttetést meg kell szakítani arra az időre, amíg zavarimpulzusok lépnek fel.
- Az összeköttetés csak akkor álljon helyre, ha az adatok stabilak
- Az FF1-en keresztül visszacsatolt jelet legalább a KH1 kombinációs hálózat legnagyobb jelkésleltetési idejével ( $t_2$ ) késleltetni kell.

# Szekvenciális hálózatok

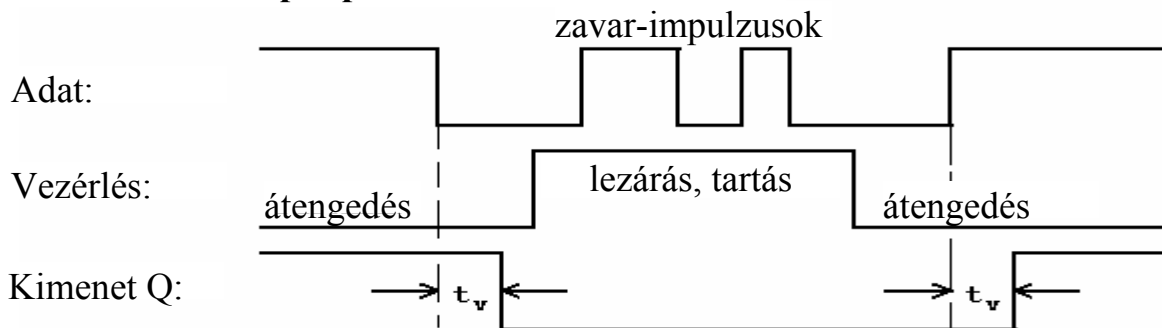
## Alap-Flipflop

A flipflop fenti jellemző felhasználási példájából levezethető **alapvető** időbeli és logikai **flipflop-tulajdonságok**:

### Latch-flipflop:

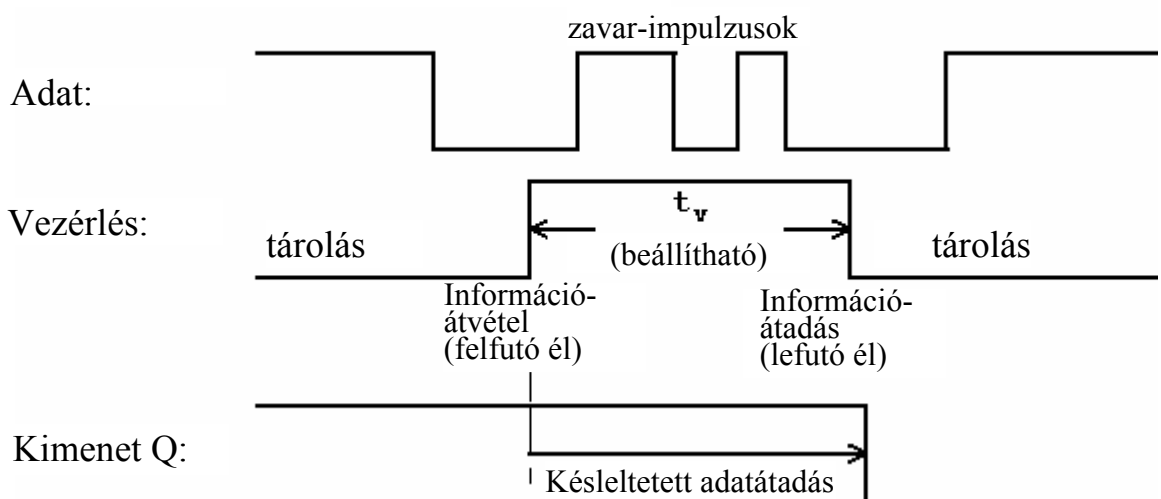


### Vezérelt latch-flipflop:



Fenti követelmények megvalósítására egyik lehetőség a

### Kétszeres él-vezérlés:



# Szekvenciális hálózatok

## Alap-Flipflop, egyszerű Latch-Flipflop tervezése

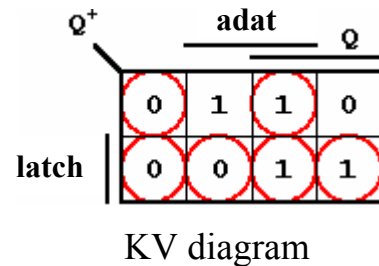
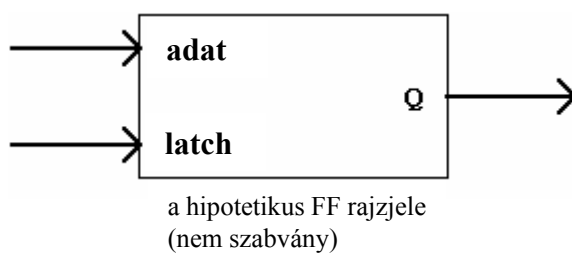
A megvalósítandó flipflop két állapota:

- „áteresztés” (unlatch):  
Az adatbemeneten lévő jel megjelenik a kimeneten.
- „lezárás, tartás” (latch):  
A közvetlenül ezen állapot beállta előtti időpontban a bemeneten meglévő jel értéke „befagy” és az állapot végéig a kimeneten rendelkezésre áll.

adatbemenet	latch vezérlőbem.	$Q^+$
d	0 (unlatch)	d
x	1 (latch)	Q

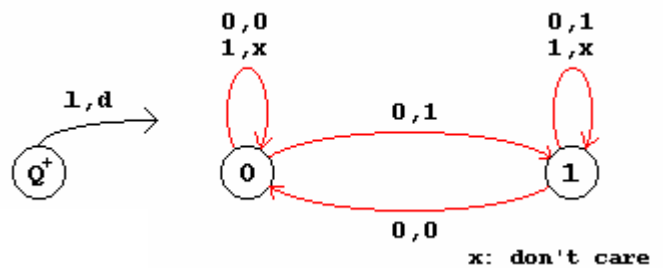
(d = adat)

(x = don't care)

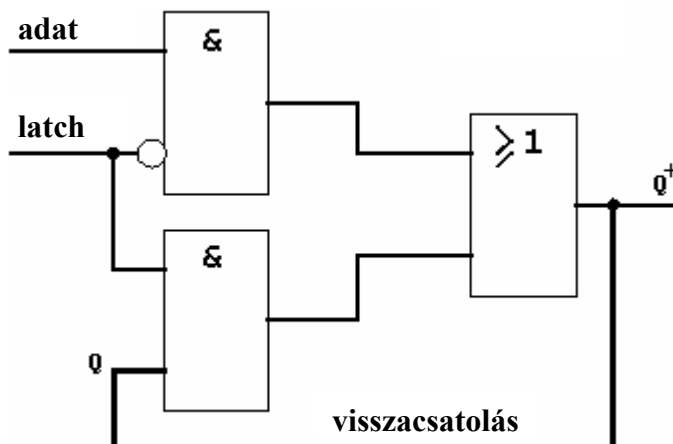


$$Q^+ = \overline{\text{latch}} \cdot \text{adat} \vee \text{latch} \cdot Q$$

Stabil állapotokra érvényes:  $Q^+ = Q$



Állapotdiagram  
l: latch, d: adat



Kapuarámkörös megvalósítás

# Szekvenciális hálózatok

## RS-”alap”flipflop

Alapvető flipflop-tulajdonságok:

- bistabil működés, a két stabil állapot a „0” és az „1”
- beállított érték (állapot) tárolása

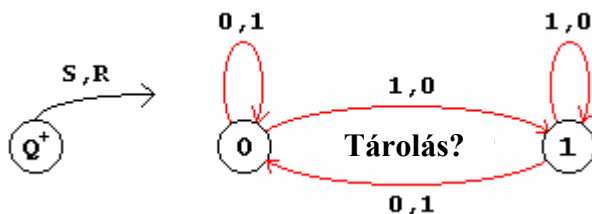
(az élvezérlés nem alapvető tulajdonság!), tehát három funkciót kell megvalósítani.

A két folyamat elnevezése, melyek a flipflop beállításához, azaz a megőrzendő

logikai érték felvételéhez vezetnek: **beírás** (*set*) és **törlés** (*reset*)

S	R	funkció	Q <sup>+</sup>
1	0	Set	1
0	1	Reset	0
?	?	Tárolás	Q

RS-flipflop általános igazságtáblázata:  
(S és R bemenetek „vezérlik” a FF-ot.)



állapotdiagram

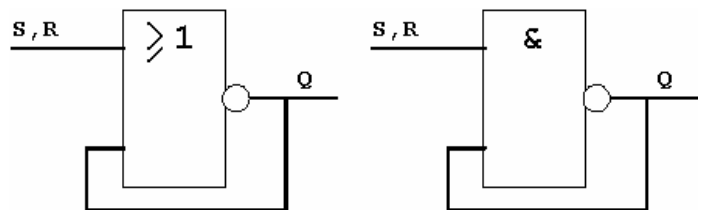
NOR ill. NAND megvalósítás

**kiindulási alapja:**

A visszacsatolt jel dominanciája biztosítja a tárolás-funkciót.

NOR: S=R=„0”

NAND: S=R=„1”



Q <sup>+</sup>	R		S
	0	0	d
	0	0	1
Q	1	0	d
	1	0	1
	tárolás	törlés	beírás

A mennyiben a „don't care”-ket d=0 -val helyettesítjük be, akkor a függvényegyenlet:

$$Q^+ = \bar{R} \cdot (S + Q) = R + \overline{\overline{S + Q}}$$

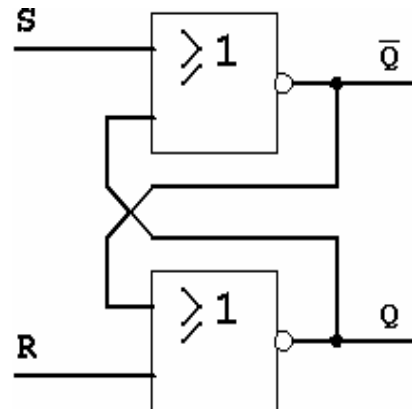
# Szekvenciális hálózatok

## RS-”alap”flipflop

RS-flipflop NOR megvalósítása:

$$Q^+ = \overline{R} \cdot (S + Q) = R + \overline{\overline{S + Q}}$$

		R		S
Q <sup>+</sup>		0	0	1
Q		1	0	1
		tárolás	törlés	beírás



A KV diagramban bekarikázott állapotok a stabil állapotok

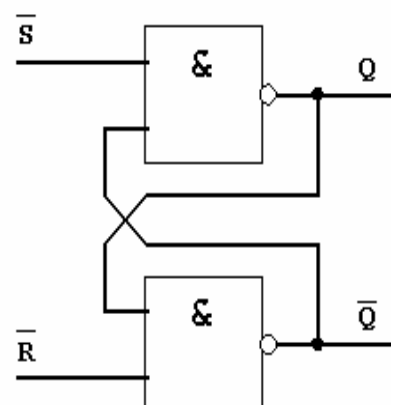
Amíg a NOR-megvalósításnál az RS-flipflop R=S=„0” bemeneteknél tárol, addig a NAND-nál R=S=„1” fennállásakor. (Ezen bemeneti értékek fennállásakor csak a domináns visszacsatolások érvényesülnek.)

- A bemeneti-kimeneti csatlakozások és az előzőekben definiált R, S és Q egymáshoz rendelése dönti el, hogy a vezérlés „active high” vagy „active low” lesz-e.
- Ezen alap-FF-kapcsolásnál nem egyértelmű, hogy a kimenet melyik állapotba áll be akkor, ha a FF az R=S=„1” állapotból az R=S=„0” állapotba megy át (a két bemenet „1”-ről egyszerre vált „0”-ra). Ebben az esetben a végállapot a „véletlenszerű” kapu-késleletelési időktől függ.  
→ „döntő verseny” a különböző futamidejű jelek között (race)
- Az az elvárható, de nem feltétlenül szükséges tulajdonság, hogy a két kimenet egymásnak az inverze legyen, R=S=„1” esetében nem áll fenn.

Két utóbbi megjegyzés miatt az RS-NOR flipflopnál az R=S=„1” bemeneti konfiguráció nem megengedett. Ugyanez érvényes az RS-NAND FF-ra az R=S=„0” bemeneti konfigurációra.

**NAND:**

NOR-FF			Funkció	NAND-FF		
S	R	Q <sup>+</sup>				Q <sup>+</sup>
1	0	1	<b>Beírás</b>	0	1	1
0	1	0	<b>Törlés</b>	1	0	0
0	0	Q	<b>Tárolás</b>	1	1	Q
1	1	-	<b>(tiltott)</b>	0	0	-



---

# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok

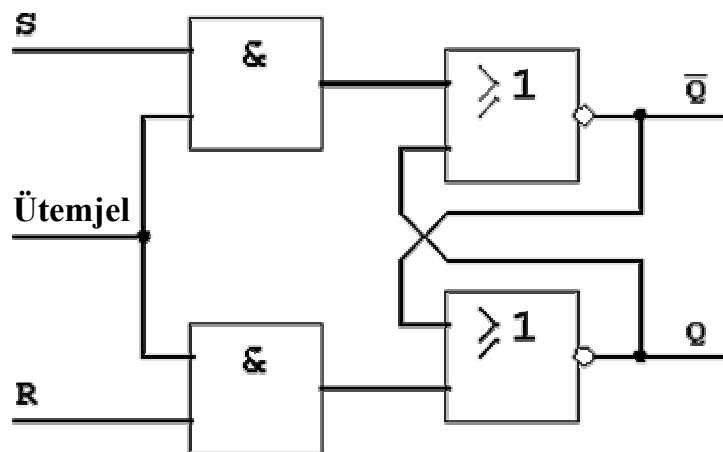
## RS-flipflop (gated latch)

---

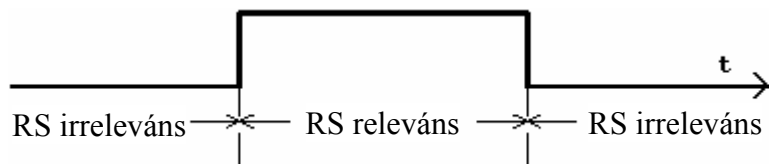
Míg az egyszerű RS-flipflop a bemenetire érkező jel-változást bármely időpillanatban, azonnal kiértékeli, addig az **ütemjel-vezérelt** flipflop (*gated latch*) csak megszabott időben képes az RS logikai bemenetek tartalmát kiértékelni. Ezt a már az előzőekben megismert latch-funkciót egy újabb vezérlő jel, az ütemjel (*clock*) teszi lehetővé.

- Ütemjel = „1”: az RS-bemenetek kiértékelődnek
- Ütemjel = „0”: az RS-bemenetek nem értékelődnek ki

Gated latch  
RS-flipflop  
kapcsolási rajza:



Ütemjel:



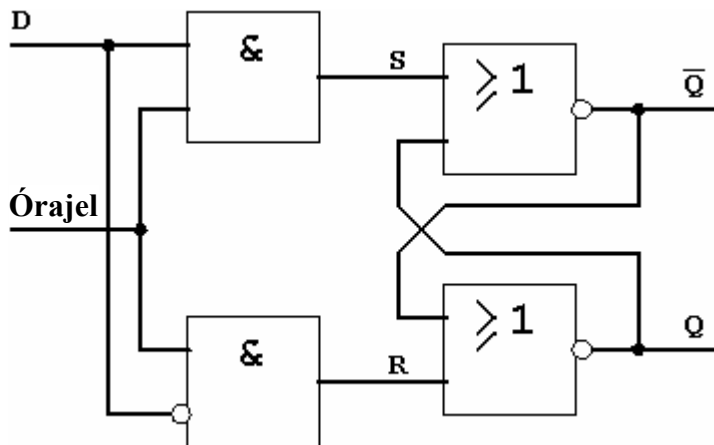
Gated latch: ütemjelszint-vezérlés, jelen esetben az ütemjel „1”-szintje az aktív, csak ekkor releváns (meghatározó) az RS-bemenet



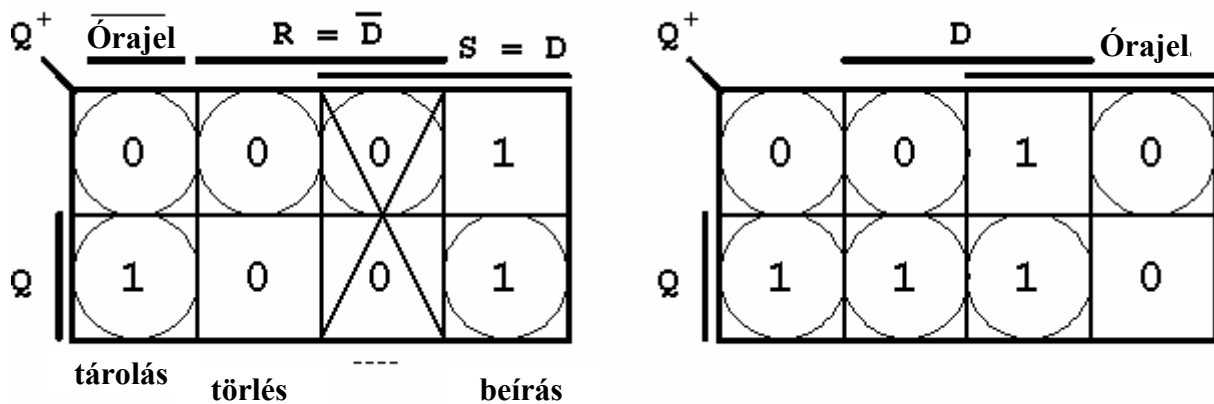
# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok

## D-flipflop, (data latch)

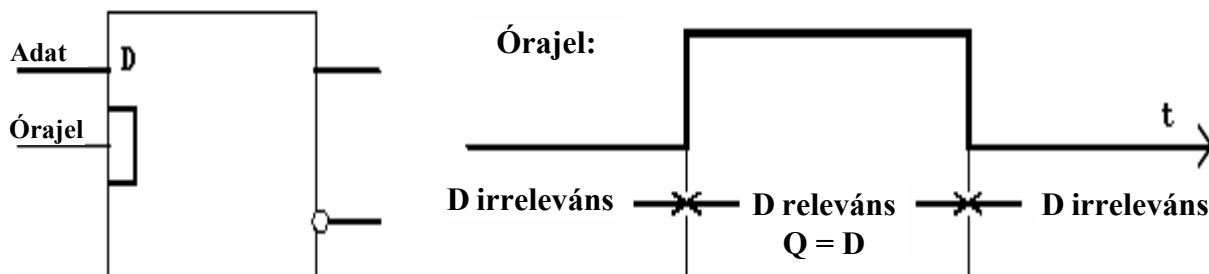
Az órajelszint-vezérelt RS-flipflop csekély változtatásával kiküszöbölhető a „tiltott” állapot ( $R=S=„1”$ ) problémája:



Az R-bemenet az invertált S-bemenet jelét kapja meg, tehát csak egy logikai bemeneti jel marad, melyet D-nek nevezünk (*data* vagy *delay*).  $R=S$  már nem lehetséges, a tárolási állapotot az ütemjel=„0” valósítja meg.



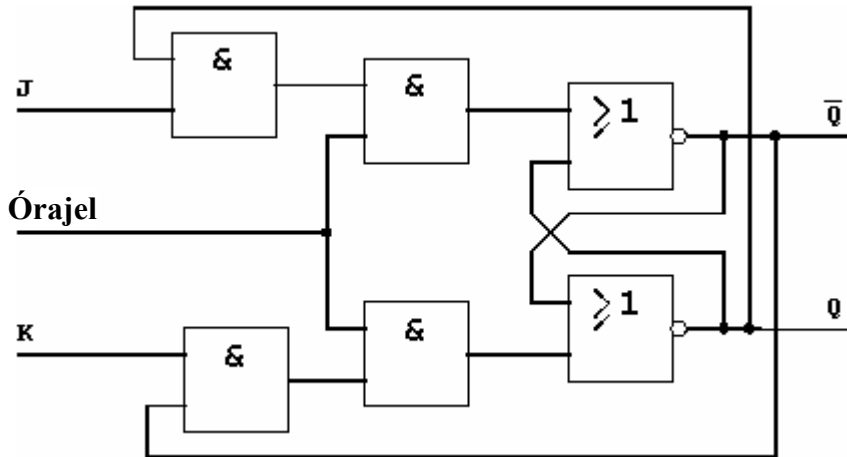
A szint-vezérlés rajzi jelölése négyszöggel:



# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok

## JK-flipflop

Az RS-flipflop tiltott állapotának elkerülésére másik lehetőséget egy újabb, negyedik funkció bevezetése ad. Ez pedig az „átváltás”-funkció (angolul *toggle*), melyet a kimenetek újabb visszacsatolásával lehet megvalósítani.



Az új flipflop logikai bemeneteit J és K jelöli (J = jump, K = kill).

$Q^+$	Órajel	K (R)		J (S)
	0	0	1	1
$Q$	1	0	0	1
	tárolás	törlés	átváltás	beírás

**Igazságtáblázat** (az órajel-periódus alatt a JK-bemenetek nem változnak):

J	K	$Q^+$	Funkció
1	0	1	Beírás (set)
1	1	$\bar{Q}$	Átváltás (toggle)
0	0	Q	Tárolás
0	1	0	Törlés (reset)

**Állapotváltási táblázat:**

Algebrai egyenlet:

$$Q^+ = \bar{K}Q \vee J\bar{Q}$$

$Q \rightarrow Q^+$	J	K	Funkció	Term
0 → 0	0	x	not jump	$J\bar{Q}$
0 → 1	1	x	jump	$J\bar{Q}$
1 → 0	x	1	kill	$\bar{K}Q$
1 → 1	x	0	not kill	$\bar{K}Q$

# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok Élvezérlés

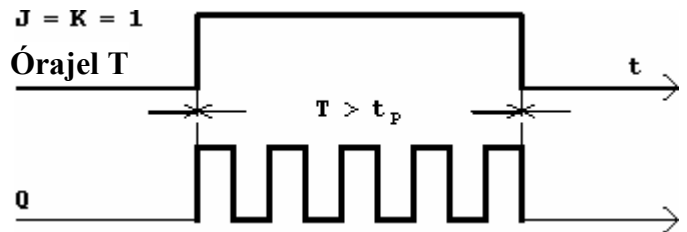
A JK-flipflop „átváltás” funkciójának bevezetésével a szintvezérlés problémákhoz vezet:

Például: legyen  $J = K = „1”$  és  $Q = „0”$

Az aktív órajel-szint ráadásakor  $t_p$  késleltetési idő eltelte után előálló új helyzet:

$J = K = „1”$  és  $Q = „1”$

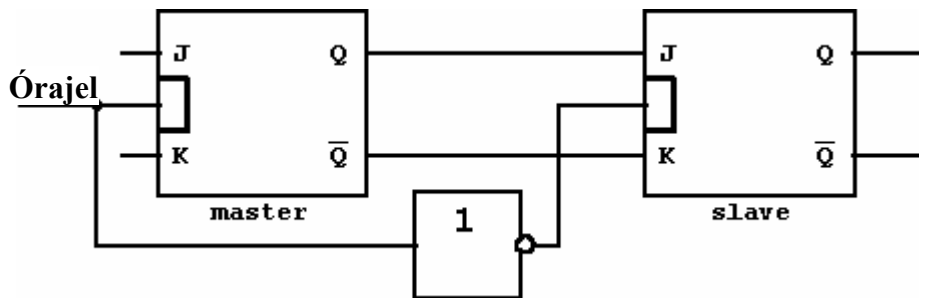
→ Nyitott órajel-szint fennállásakor a folyamat ismétlődik → Q oszcillál (*race around*)



Órajel-impulzus szélessége lehessen bármekkora, de csak egyszeri állapotváltás történjen

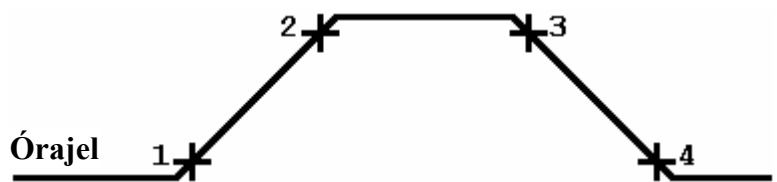
→ élvezérlés

**Master-Slave megoldás (JK-flipfloppal):**



**Master-Slave FF működése:**

1. *Master* és *Slave* elválasztása
2. JK-jelek bevitele a *Master*-be
3. JK-bemenetek lezárása
4. Adatok átadása a *Master*-ból a *Slave*-be



Az adott kapcsolás (Master-Slave) esetében az adat átvétele a felfutó ( $0 \rightarrow 1$ ), és a Q kimenet változása a lefutó ( $1 \rightarrow 0$ ) órajel-él hatására következik be.

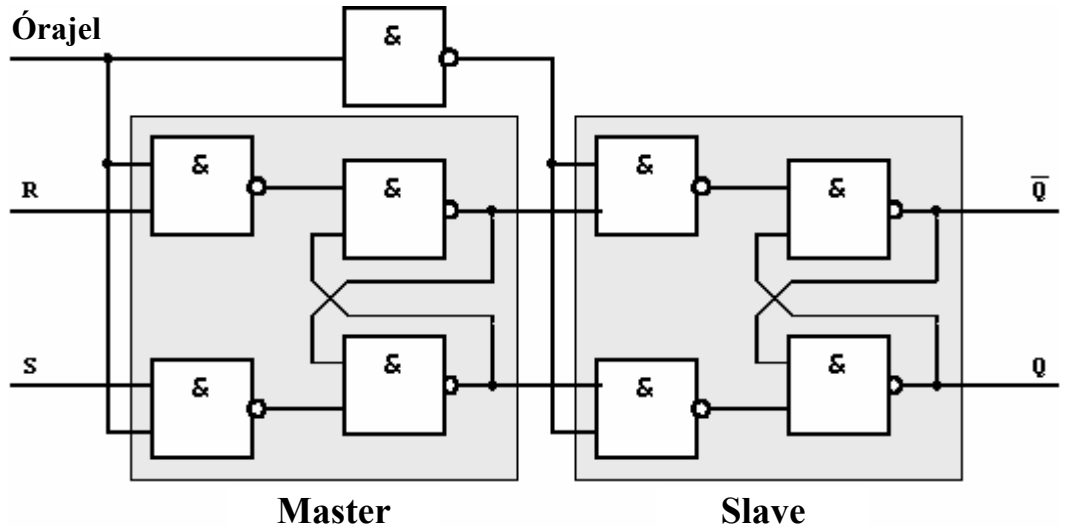
Azon flipflop-okat, melyek állapotot csak az egyik órajel-él hatására váltanak → **egyél-vezérelt fliflop**-nak nevezzük:

- Felfutó él-vezérelte flipflop
- Lefutó él-vezérelte flipflop

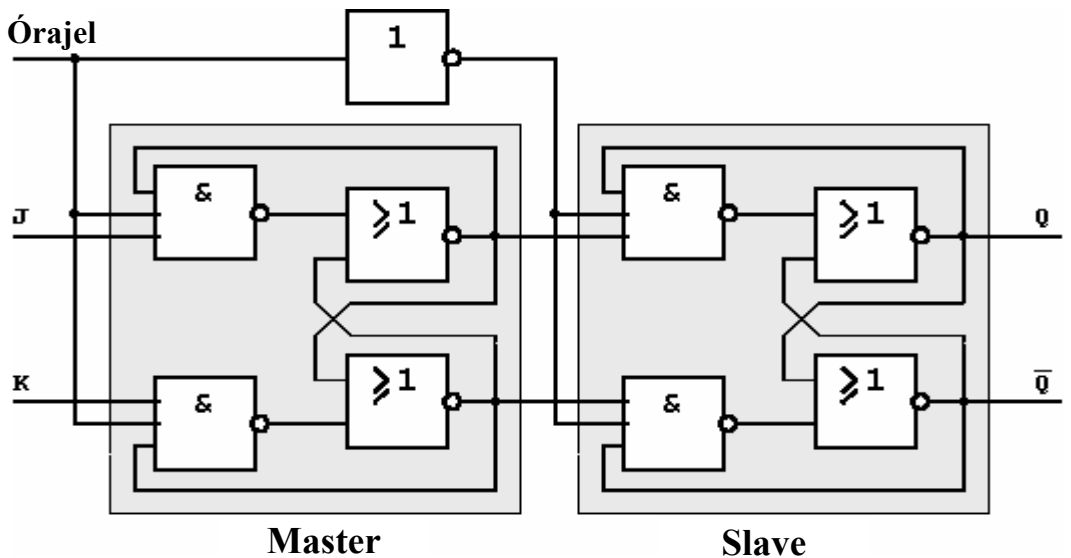
Az információ átvételét és kiadását ugyanaz az órajel-él váltja ki.

# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok Élvezérlés

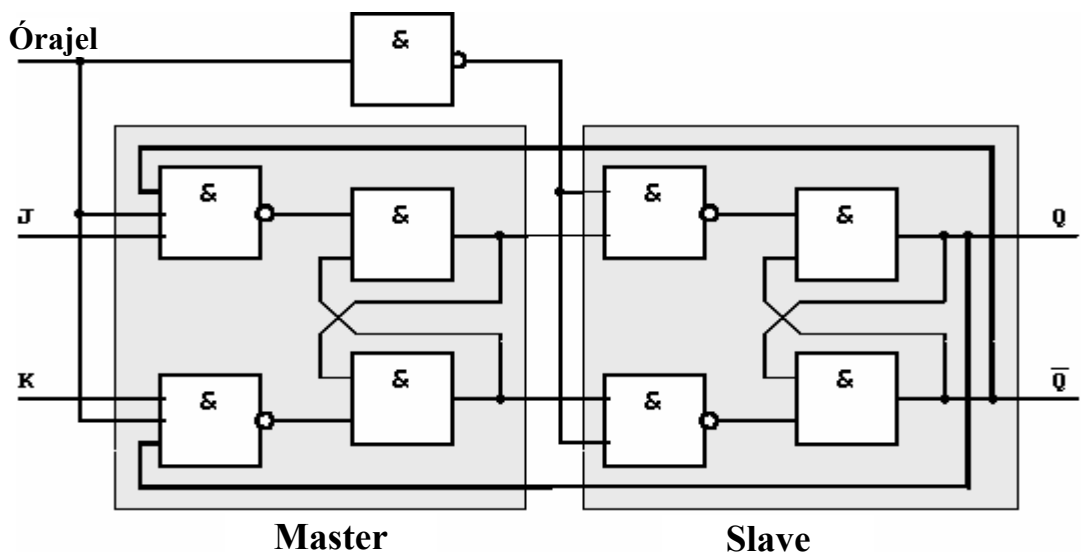
Élvezérelt  
master-  
slave-  
flipflop RS-  
flipflopból  
felépítve:



Élvezérelt  
master-  
slave-  
flipflop JK-  
flipflopból  
felépítve:



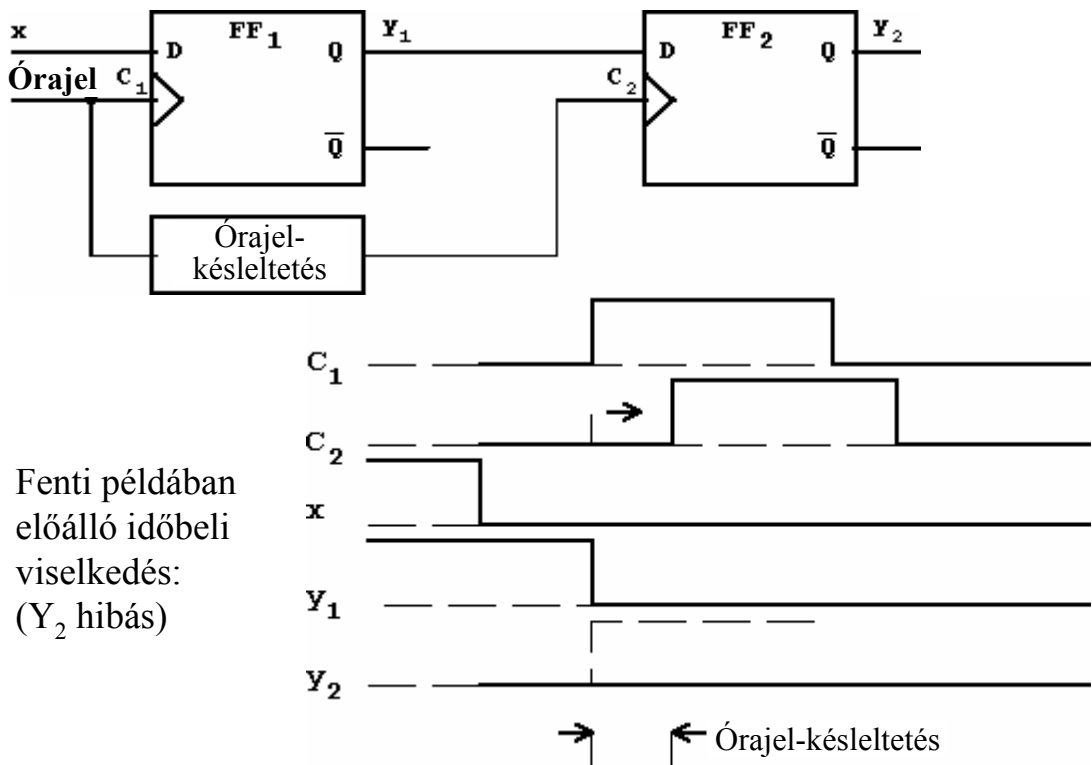
Élvezérelt  
master-  
slave-  
flipflop  
leggyakoribb  
változata



# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok Kétél-vezérlés

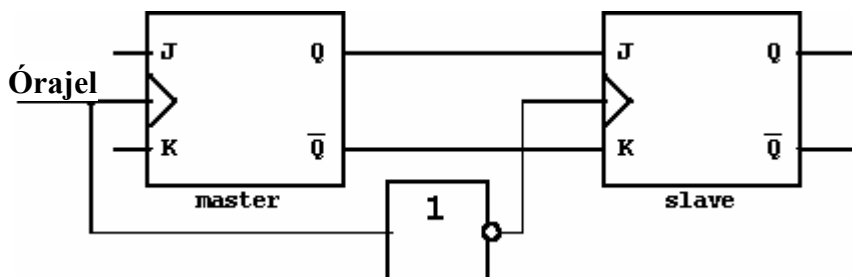
Nagy kiterjedésű, összetett kapcsolásoknál többszörösen elágaztatott órajel-vezeték esetében több flipflop egyidejű (szinkron) ütemezése függ az órajel időbeli terjedési viszonyaitól is.

Például:



Fenti példában előálló időbeli viselkedés: (Y<sub>2</sub> hibás)

Fenti probléma elkerülhető az un. „**kétél-vezérelt flipflop**”-ok használatával. Két darab egyél-vezérelt MS-flipflop Master-Slave konfigurációban:



Az információ az órajel felfutó élével kerül be a *master*-be, ahol tárolódik a lefutó élig, és csak a lefutó él bekövetkeztékor kerül rá a kimenetre (*slave*-be).

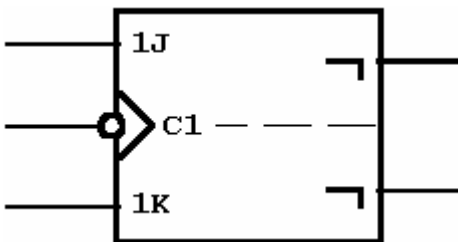
előforduló max. ütemjel-késleltetés  $\leq$  felfutó és lefutó él közötti különbség

# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok

## Kétél-vezérlés

### Kétél-vezérelt JK-flipflop:

Információ átvétele az órajel felfutó élével, kiadása a kimenetre a lefutó éllel.

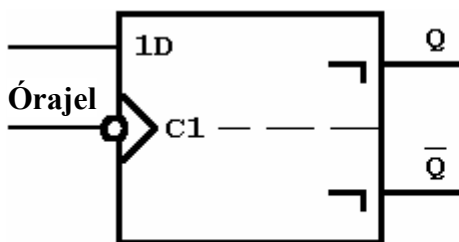


Rajzi jelek:	
1J,1K	Az 1-es bemenet függ a C1 időbeliségétől
	aktív: lefutó órajel-él
	"késleltetett" kimenet

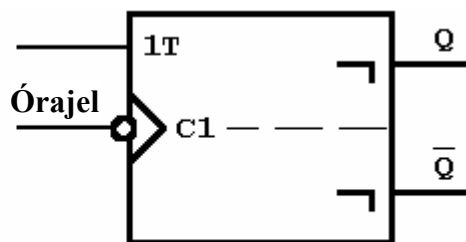
# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok

## D-flipflop, T-flipflop

Az élvezérelt JK-flipflopból funkciói szerint levezethető flipflop-típusok közül a D-flipflop és a T-flipflop rendelkezik gyakorlati jelentőséggel.



D-Flipflop



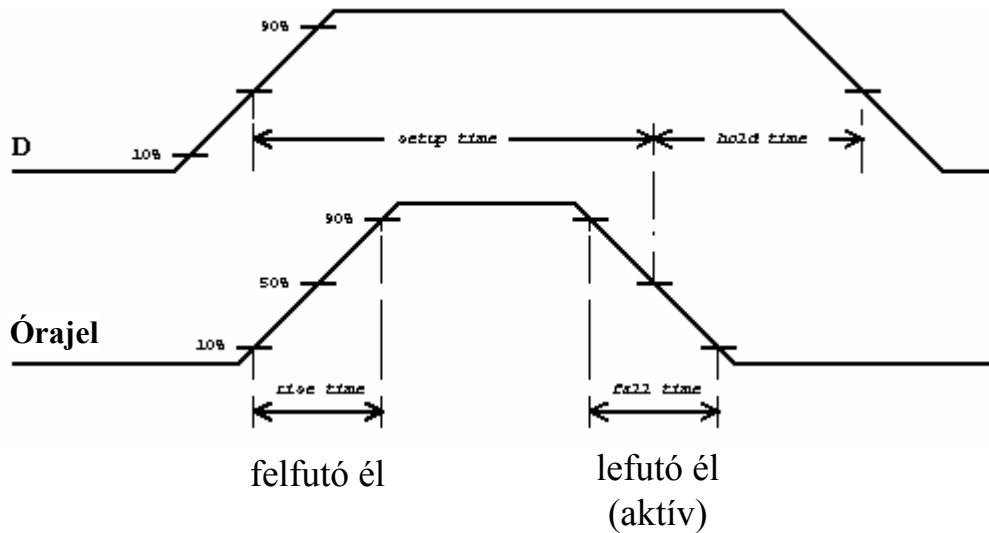
T-Flipflop

D-Flipflop		
D	Q <sup>+</sup>	Funkció
0	0	adat-átvétel
1	1	adat-átvétel

T-Flipflop		
T	Q <sup>+</sup>	Funkció
0	Q	tárolás
1	$\bar{Q}$	toggle

# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok D-flipflop, T-flipflop

Információ-átvétel alapvető időbeli követelményei:



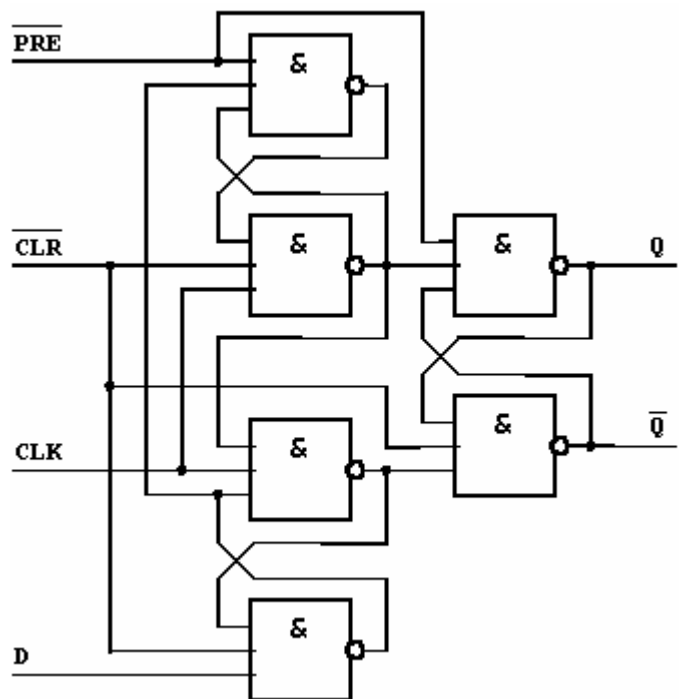
A 74LS-áramkörcsalád jellemző  
setup-time és hold-time értékei:

$$t_{SU} > 20 \text{ ns}$$
$$t_H > 5 \text{ ns}$$

## D-flipflop SN 7474

Az integrált flipflop-oknál  
többnyire két új aszinkron  
vezérlő bemenet van:

$\overline{\text{PRE}}$  - preset  
 $\overline{\text{CLR}}$  - clear (active low)



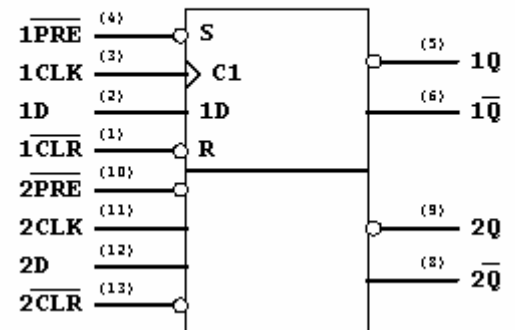
# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok D-flipflop, T-flipflop

## D-flipflop SN 7474

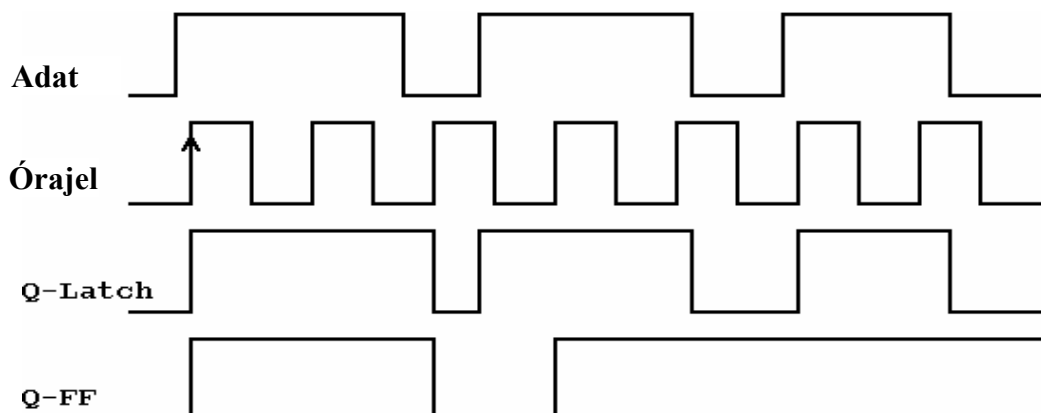
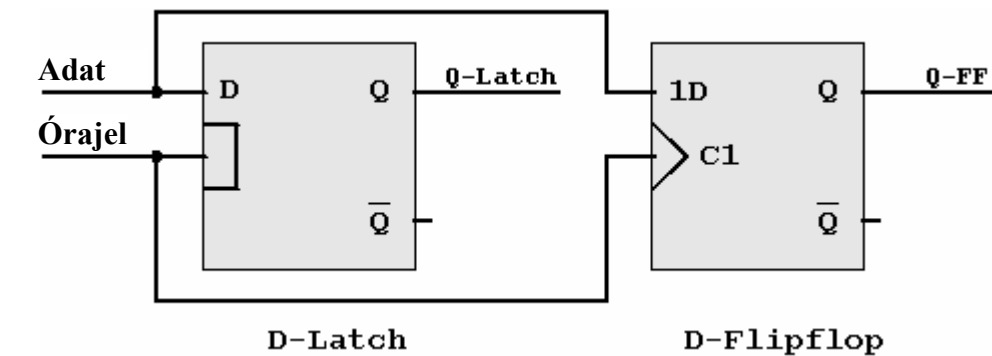
bemenetek				kimenetek	
$\overline{Preset}$ PRE	$\overline{Clear}$ CLR	órajel CLK	adat D	$Q^+$	$\overline{Q}^+$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H		H	H	L
H	H		L	L	H
H	H	L	X	Q	$\overline{Q}$

74LS74:

Kettős, felfutó évezérelt D-flipflop ,  
preset és clear vezérlő bemenetekkel



## Latch/flipflop közötti különbség:





# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok

## Áttekintő táblázat

		Flipflop-típus			
órajel	vezérlés	RS	JK	D	T (adatbem. nincs)
nincs	szint- vezérlés				
nincs	él- vezérlés				
van	egyszeres szintvezérlés				
van	kétszeres szintvezérlés				
van	egyszeres élvezérlés				
van	kétszeres élvezérlés				

órajel-bemenet	rajzi jel
Szintvezérelt órajel-bemenet. Azon bemeneti változók, melyek C-től függenek, C=1 esetében hatásosak.	
Élvezérelt órajel-bemenet. Azon bemeneti változók, melyek C-től függenek, C órajel 0 → 1 átmenetekor lesznek hatásosak (felfutó él).	
Élvezérelt órajel-bemenet. Azon bemeneti változók, melyek C-től függenek, C órajel 1 → 0 átmenetekor lesznek hatásosak (lefutó él).	

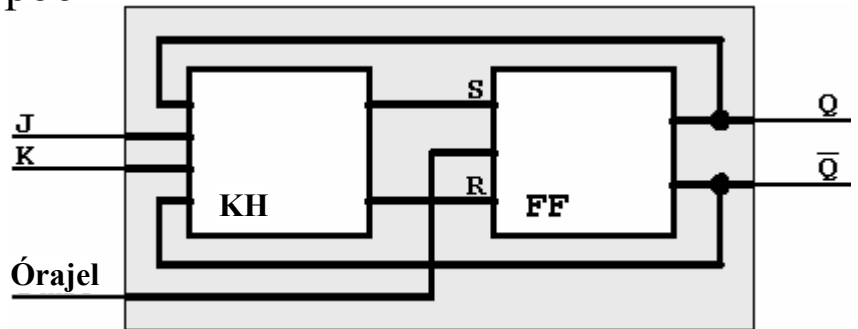
# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok

## Flipflop-átalakítás

**Példa:** RS-flipflopból  
JK-flipflop

**Kritérium:**

Átváltás során az  
ütemjel-vezérlés  
nem változhat



Az RS-flipflop és JK-flipflop átmeneti táblázatai:

JK-Flipflop			RS-Flipflop		
$Q \rightarrow Q^+$	J	K	$Q \rightarrow Q^+$	S	R
0 → 0	0	x	0 → 0	0	x
0 → 1	1	x	0 → 1	1	0
1 → 0	x	1	1 → 0	0	1
1 → 1	x	0	1 → 1	x	0

↖  
Rövidített változat

→  
Részletes változat

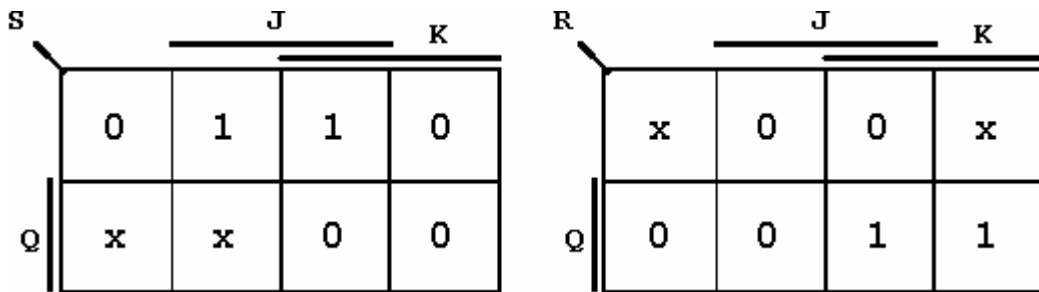
$Q \rightarrow Q^+$	J	K	S	R
0 → 0	0	0	0	x
0 → 0	0	1	0	x
0 → 1	1	0	1	0
0 → 1	1	1	1	0
1 → 0	0	1	0	1
1 → 0	1	1	0	1
1 → 1	0	0	x	0
1 → 1	1	0	x	0

# Szekvenciális hálózatok, Ütemjel-vezérelt flipflop-ok

## Flipflop-átalakítás

### Példa: RS-flipfloból JK-flipflop (folytatás)

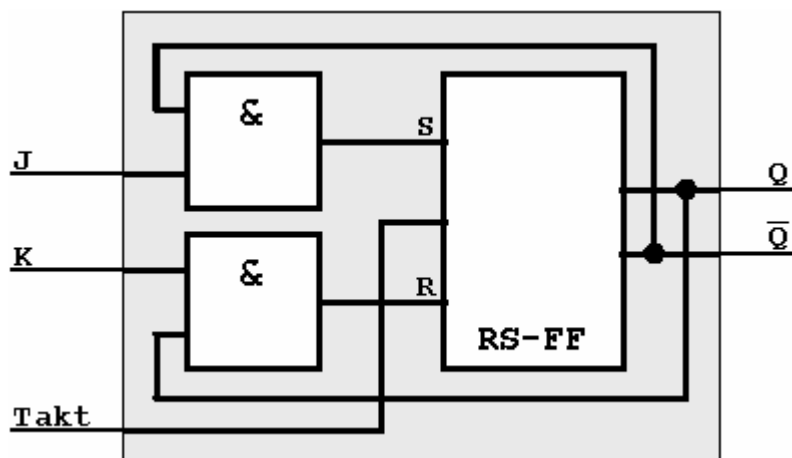
A JK/RS kombinációs hálózat meghatározásához szükséges KV-diagram:



Egyszerűsítés után adódó S és R függvények algebrai alakja

$$S = J \wedge \overline{Q} \quad R = K \wedge Q$$

Megvalósítás:



# Szekvenciális hálózatok

## Regiszterek

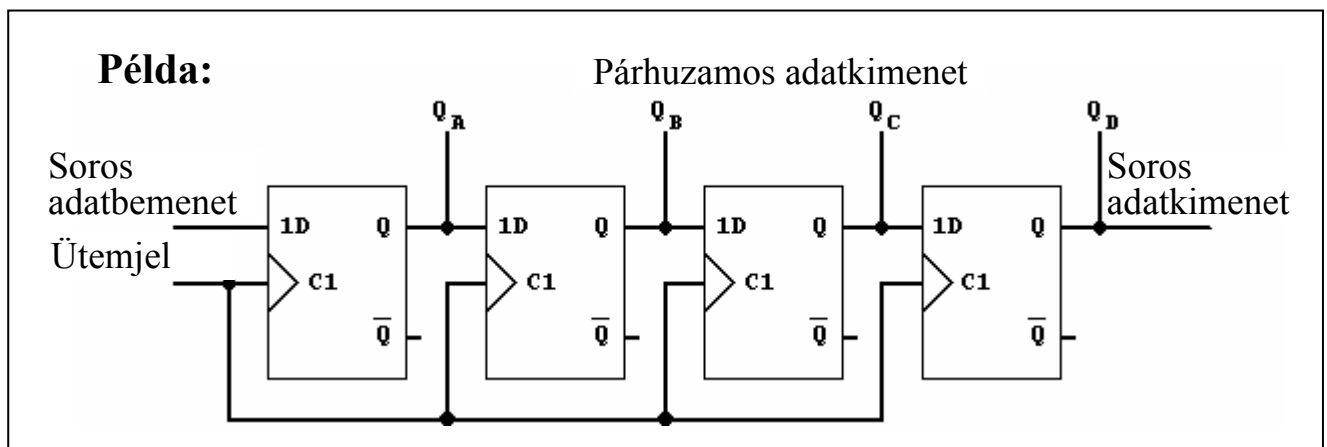
Flipflop-ok: 1 bit kapacitású információ-tárolók

Nagyobb tárolókapacitás több flipflop összekapcsolásával lehetséges, melyeket egyetlen órajel-vezeték vezérel:

→ **regiszterek** (ütemezés: bitenkénti léptetés → léptetőregiszterek)

Adatbemenet	Adatkimenet	Funkció	Példa (8 Bit)
soros	soros	SISO	
soros	párhuzamos	SIPO	
párhuzamos	soros	PISO	
párhuzamos	párhuzamos	PIPO	

Elvileg az összes tárgyalt flipflop-típus alkalmas léptetőregiszter előállítására. Leginkább elterjedtek a D- és JK-flipflop-ok (szint és élvezérelt is).



Fenti funkciókon túlmenően a léptetőregiszterek további vezérlő-bemenetekkel is rendelkeznek:

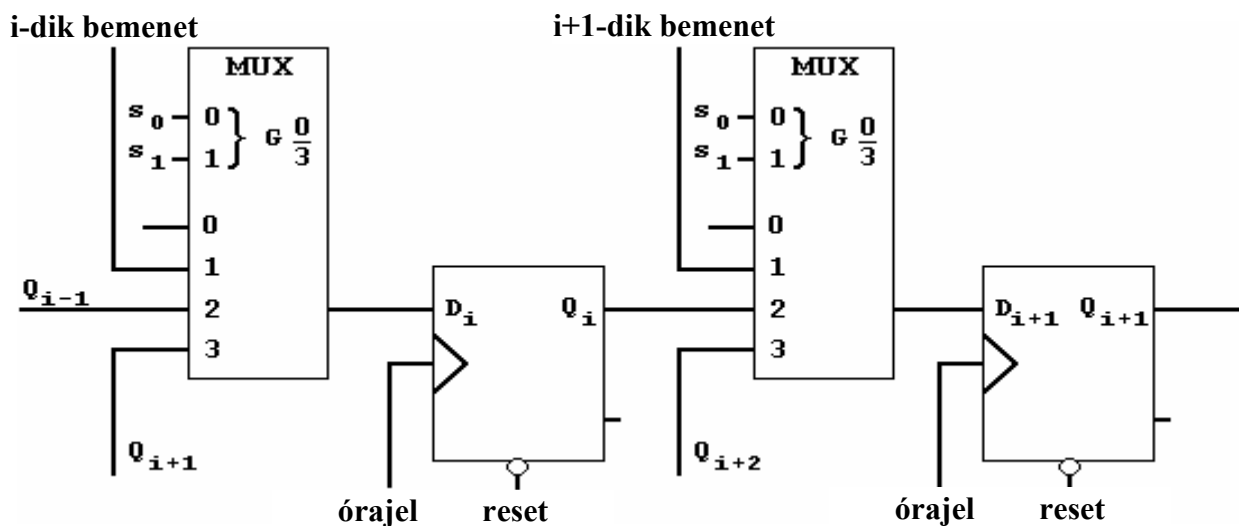
- párhuzamos adatok beírásának ill. kiadásának vezérlő bemenetei
- reset-bemenet (nulla-vektor betöltése)
- léptetési irány megadására szolgáló bemenet

# Szekvenciális hálózatok

## Regiszterek

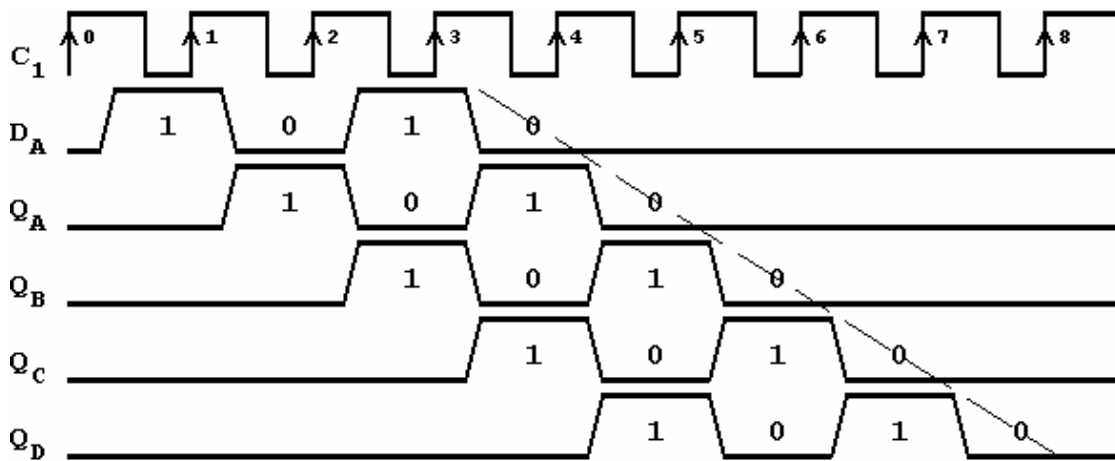
Különböző léptetőregiszter-típusokat lehet a következő kapcsolás „programozásával” előállítani:

S <sub>1</sub>	S <sub>0</sub>	Funkció
0	0	nincs funkciója
0	1	párhuzamos betöltés
1	0	jobbra-léptetés
1	1	balra-léptetés



**Példa:** (az előző oldal 4-bites léptetőregiszterére)

- legyen jobbra-léptető SISO ill. SIPO regiszter
- legyen a sorosan beírandó vektor: (1010)
- a kiindulási állapot legyen:  $D_i = Q_i = „0”$

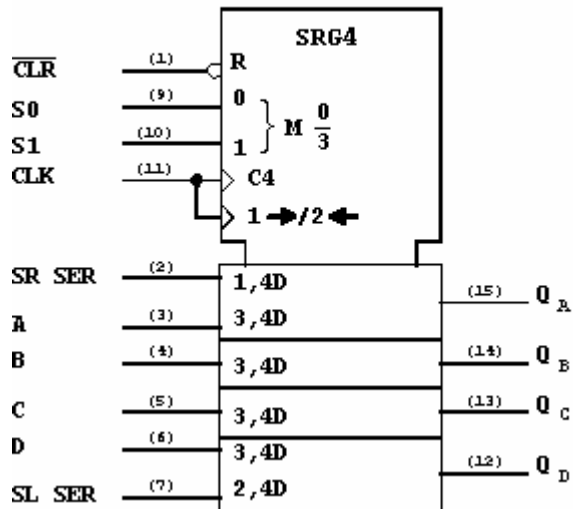


# Szekvenciális hálózatok

## Regiszterek

**Példa** integrált léptetőregiszterre:

74LS194: 4-Bit Bidirectional Universal Shift Register



74x194 igazságtáblázata:

Inputs										Outputs			
Clear	Mode		Clock	Serial		Parallel				Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>D</sub>
	S <sub>1</sub>	S <sub>0</sub>		Left	Right	A	B	C	D				
L	x	x	x	x	x	x	x	x	x	L	L	L	L
H	x	x	L	x	x	x	x	x	x	Q <sub>A0</sub>	Q <sub>B0</sub>	Q <sub>C0</sub>	Q <sub>D0</sub>
H	H	H		x	x	a	b	c	d	a	b	c	d
H	L	H		x	H	x	x	x	x	H	Q <sub>An</sub>	Q <sub>Bn</sub>	Q <sub>Cn</sub>
H	L	H		x	L	x	x	x	x	L	Q <sub>An</sub>	Q <sub>Bn</sub>	Q <sub>Cn</sub>
H	H	L		H	x	x	x	x	x	Q <sub>Bn</sub>	Q <sub>Cn</sub>	Q <sub>Dn</sub>	H
H	H	L		L	x	x	x	x	x	Q <sub>Bn</sub>	Q <sub>Cn</sub>	Q <sub>Dn</sub>	L
H	L	L	x	x	x	x	x	x	x	Q <sub>A0</sub>	Q <sub>B0</sub>	Q <sub>C0</sub>	Q <sub>D0</sub>