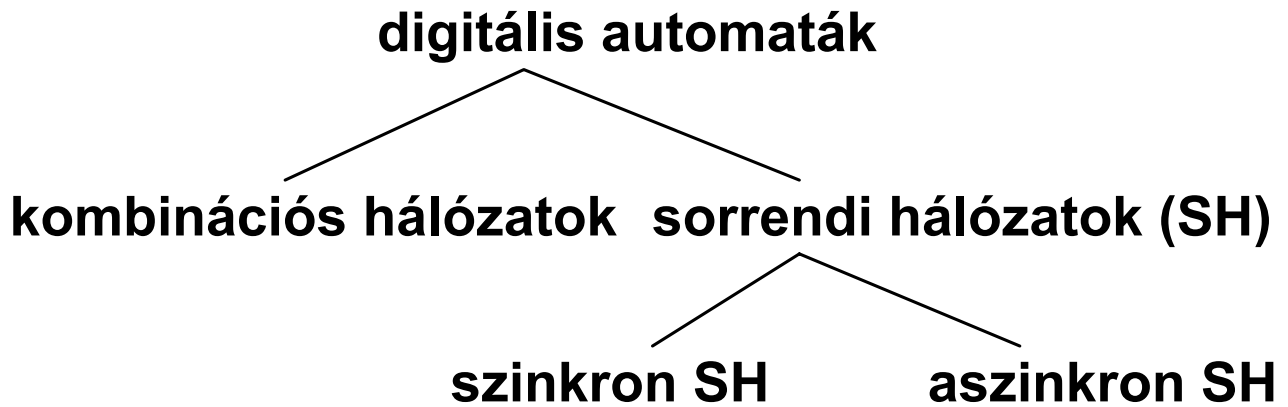


Szinkron sorrendi hálózatok tervezése

© Benesóczky Zoltán 2004

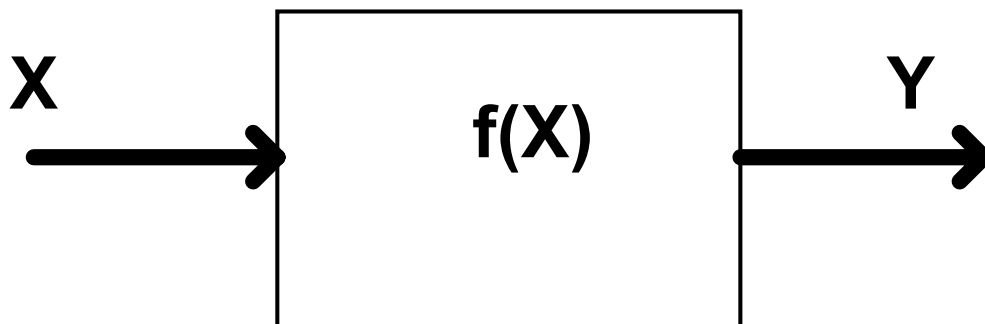
A jegyzetet a szerzői jog védi. Azt a BME hallgatói használhatják, nyomtathatják tanulás céljából. Minden egyéb felhasználáshoz a szerző beleegyezése szükséges.



Kombinációs automata

A kimenet csak az aktuális bemenettől függ.

$$Y=f(X)$$



Megvalósítása kombinációs hálózattal.

Sorrendi automata

A kimenet az *előző bemenetektől* is függ.

- *Emlékező* tulajdonsága van, melyet az *állapotokkal* (Q) reprezentálunk. Az állapotot az ún. szekunder változók kombinációja valósítja meg.

- A *következő állapot* az aktuális (t időpontbeli) állapot és az aktuális bemenet függvénye.

$$Q^{t+1}=g(X^t, Q^t),$$

- A *szinkron sorrendi hálózat (SSH) állapotait* memória tulajdonságú alkatrészek (*flip-flopok, állapot regiszter*) tárolják, egy órajellel ütemezett időpontokban.

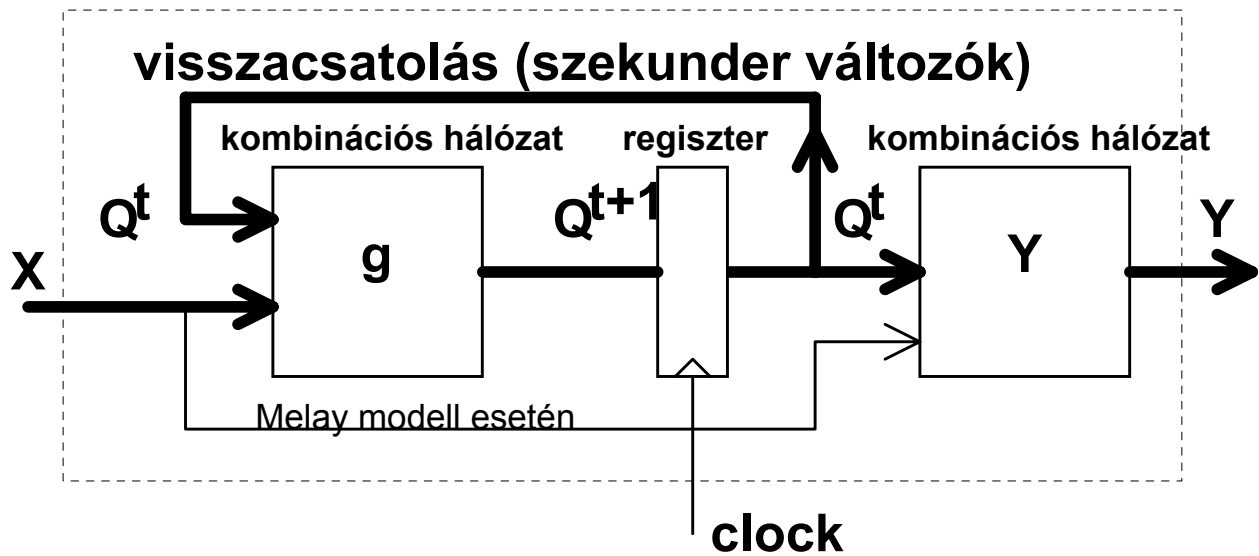
- A *kimenet* az aktuális *állapot és az aktuális bemenet függvénye* ún. *Mealy modell* szerinti működésnél.

$$Y=f(X^t, Q^t)$$

A kimenet *csak az aktuális állpottól függ* ún. *Moore modell* szerinti működés esetén.

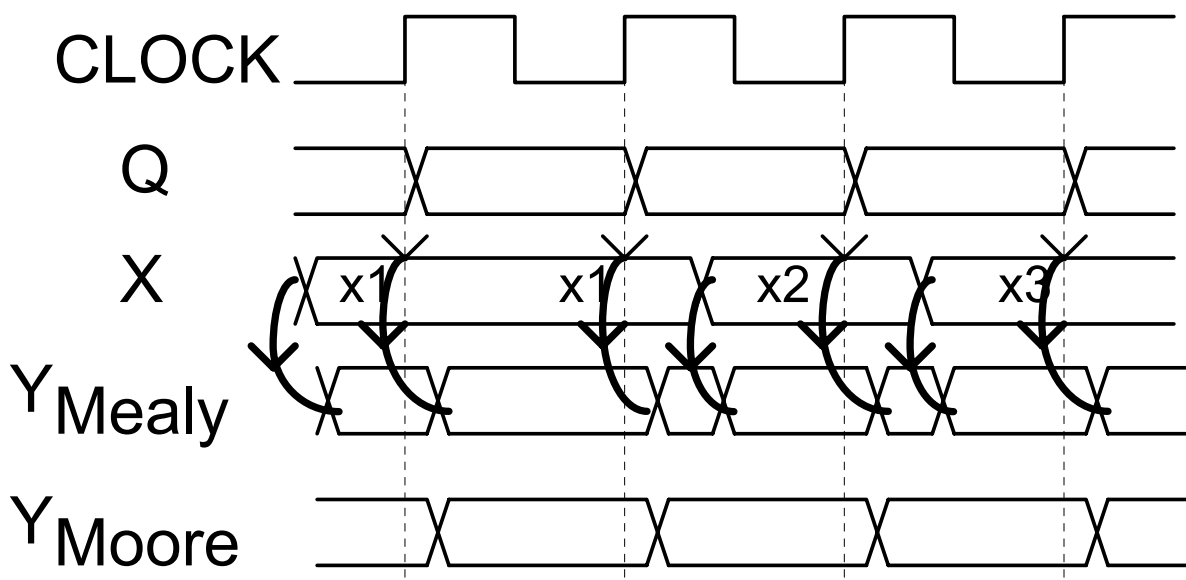
$$Y=f(Q^t)$$

A szinkron sorrendi hálózat felépítése



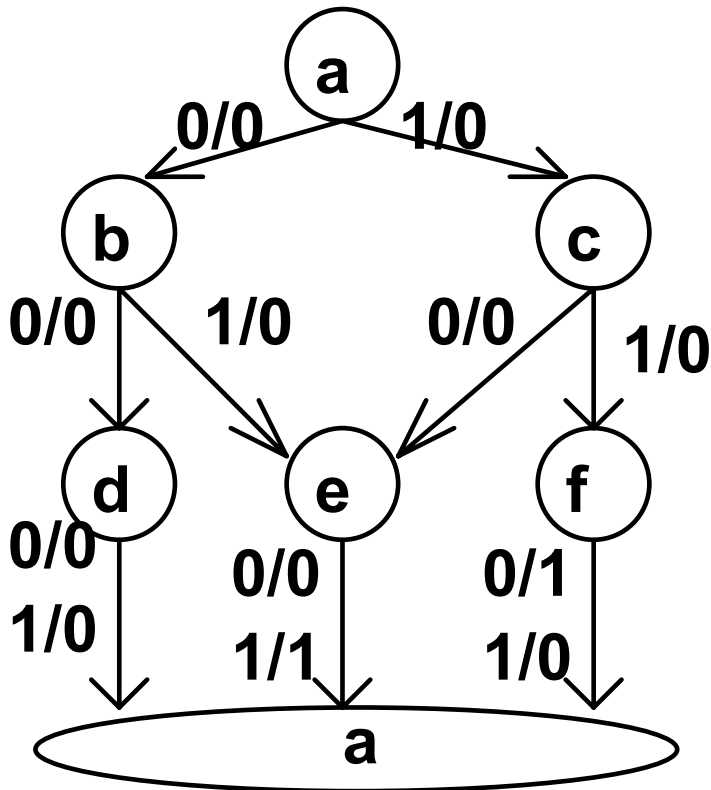
Mivel az aktuális állapot a előző állapottól és a bemenettől függ, így az a hálózatot előzőleg ért sorozattól függ. Ugyanígy a kimenet is.

A működés szemléltetése idődiagrammon



A sorrendi hálózat leírási módjai

állapotgráf



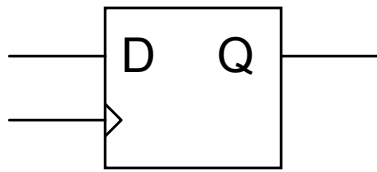
állapottábla

	x=0	x=1
a	b/0	c/0
b	d/0	e/0
c	e/0	f/0
d	a/0	a/0
e	a/0	a/1
f	a/1	a/0

- irányított gráf
- állapot -> gráfpont
- állapot átmenet ->él
- élre írva: bemeneti komb./kimenet

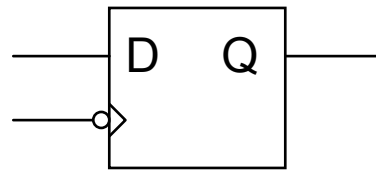
- táblázatos forma
- akt. állapot->sor
- bemeneti kombináció-> oszlop
- rubrika: köv. áll./kimenet

Legegyszerűbb állapotároló a D flip-flop.



felfutó él érzékeny

D flip-flop

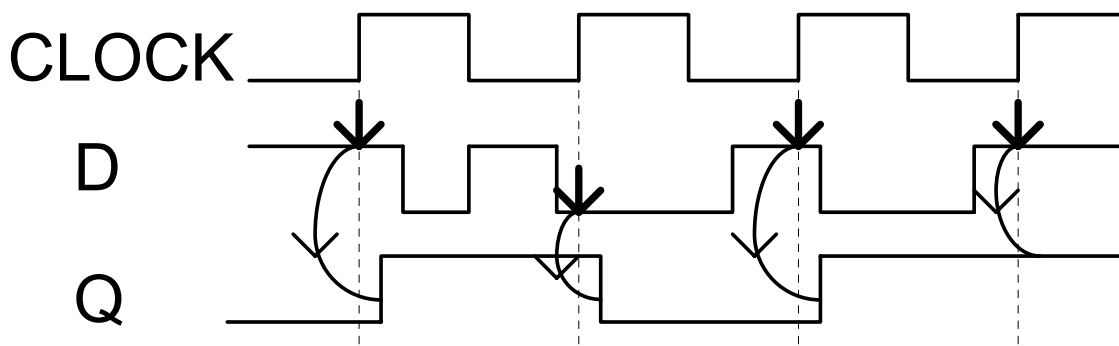


lefutó él érzékeny

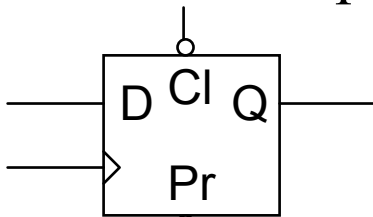
D flip-flop

Az órajel aktív élére tárolja a D bemeneten levő értéket.
 $Q^{t+1} = D$

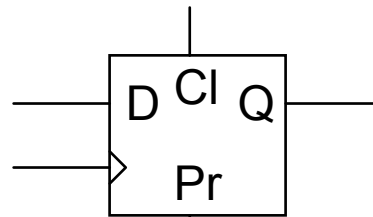
A működést szemléltető idődiagram:



Aszinkron törlő és preset bemenete is lehet:



Alacsony (Low) aktív Cl és Pr

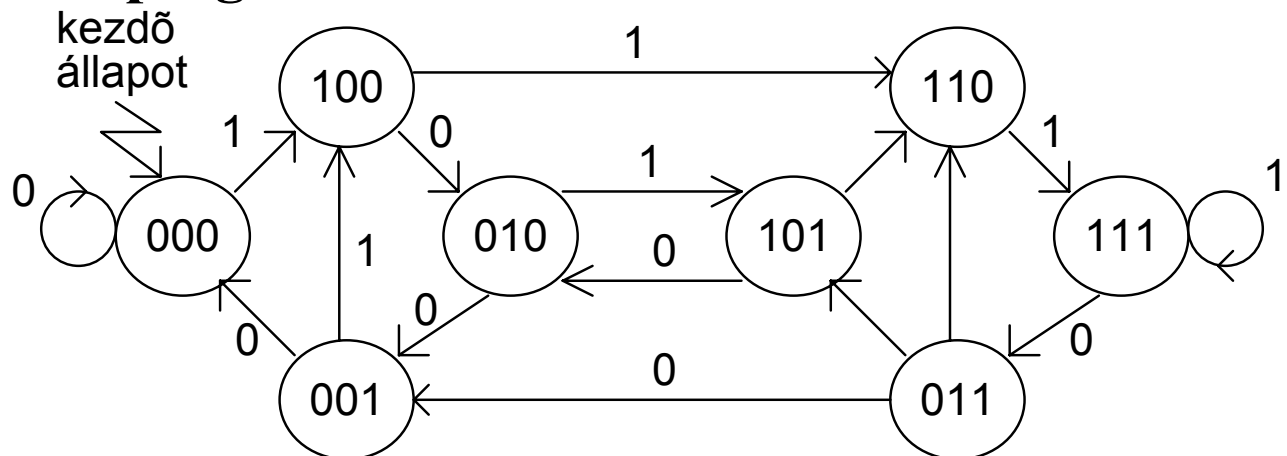


Magas (Hi) aktív Cl és Pr

Aszinkron Pr és Cl esetén kimenet az órajeltől függetlenül 1 lesz Pr, 0 lesz Cl hatására.

Készítsünk olyan áramkört, mely megjegyzi az x bemenetére az órajellel szinkronban jutó utolsó 3 értéket és azokat kiadja a kimenetein $x^{t-1}, x^{t-2}, x^{t-3}$ sorrendben. Ezt az áramkört *shiftregiszternek* nevezik.

Állapotgráf:



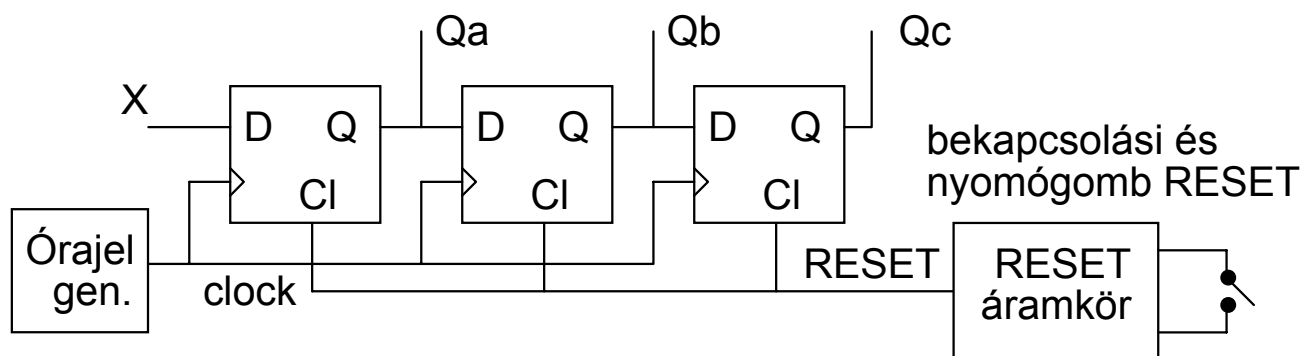
Moore modell szerinti működés, a kimenet csak az állapottól függ. A kimenet itt maga az *állapotkód*.

Kódolt állapottábla:

	$x=0$	$x=1$
000	000	100
001	000	100
010	001	101
011	001	101
100	010	110
101	010	110
110	011	111
111	011	111

- A sorrendi automata állapotait *kódolva* reprezentálja a megvalósított hálózat.
- Az állapothoz rendelt kód az *állapotkód*.
- Egy-egy *állapot kódja az állapotregiszter* (flip-flopok) kimenetének *egy meghatározott kombinációja*.
- n állapotú automata megvalósításához legalább $\lceil \log_2(n) \rceil$ darab flip-flop szükséges. (Pl. 7 állapothoz 3.)

Megvalósítás szemlélet alapján:



Az *óragerátor* állítja elő az órajelet.

A *RESET áramkör* bekapcsoláskor vagy a nyomógomb megnyomásakor impulzust ad ki, melyet a sorrendi hálózat kezdő állapotának beállítására használunk. (A továbbiakban nem rajzoljuk le külön.)

SSH tervezése szisztematikusan

Feladat:

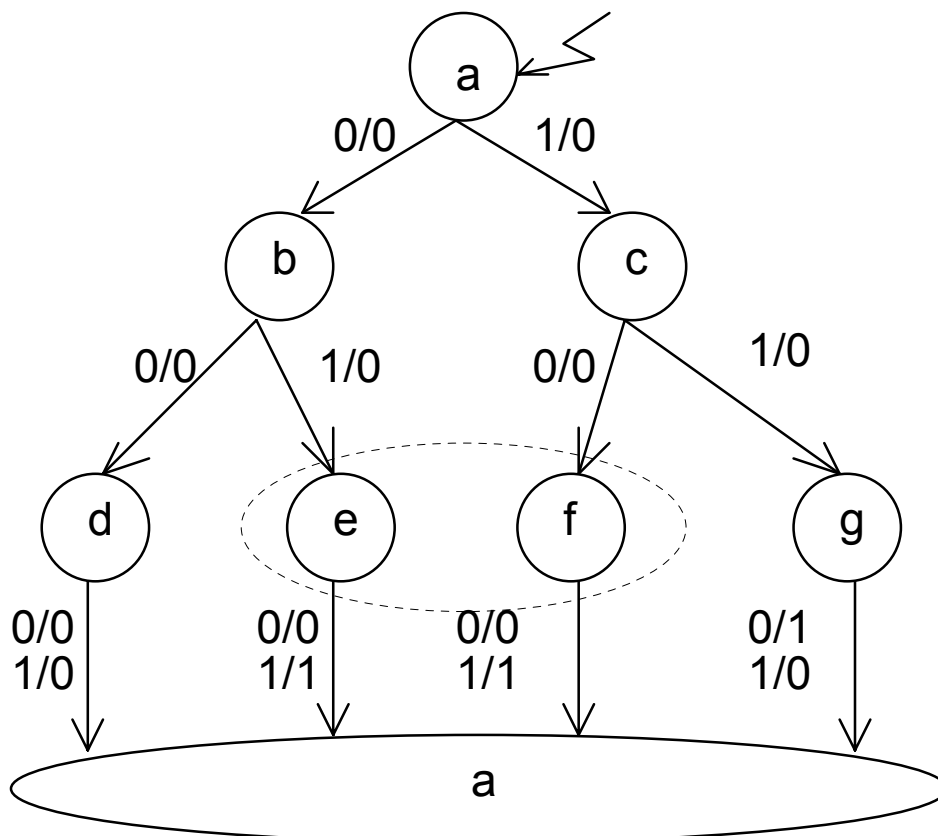
Egy sorrendi hálózat egyetlen bemenetére sorosan érkeznek 3 bites számok, az órajellel szinkronban. A megtervezendő automatának fel kell ismerni, ha a 3 bites számokban pontosan 2db 1-es van, és jelezni a 3. bit beérkezésekor.

PI:

x: 010 011 111 101 110 100 001 011

y: 000 001 000 001 001 000 000 001

1. Szöveg alapján előzetes állapotgráf megrajzolása.

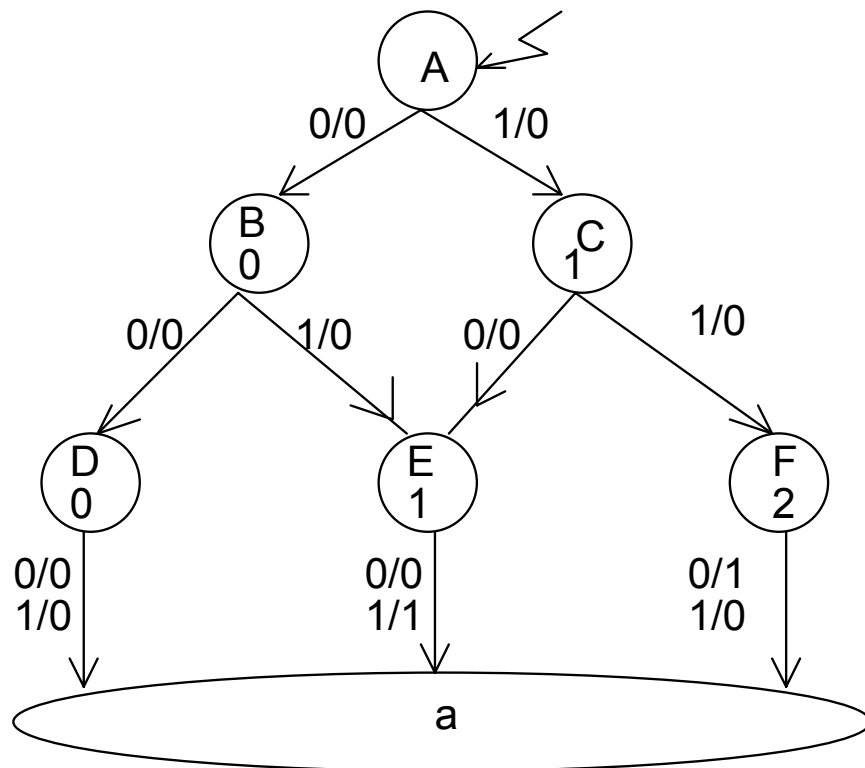


2. Állapotminimalizálás

Az előbbi e és f állapotok egyetlen állapottal helyettesíthetők, mivel csak azt kell megjegyezni, hogy hány 1-es jött.

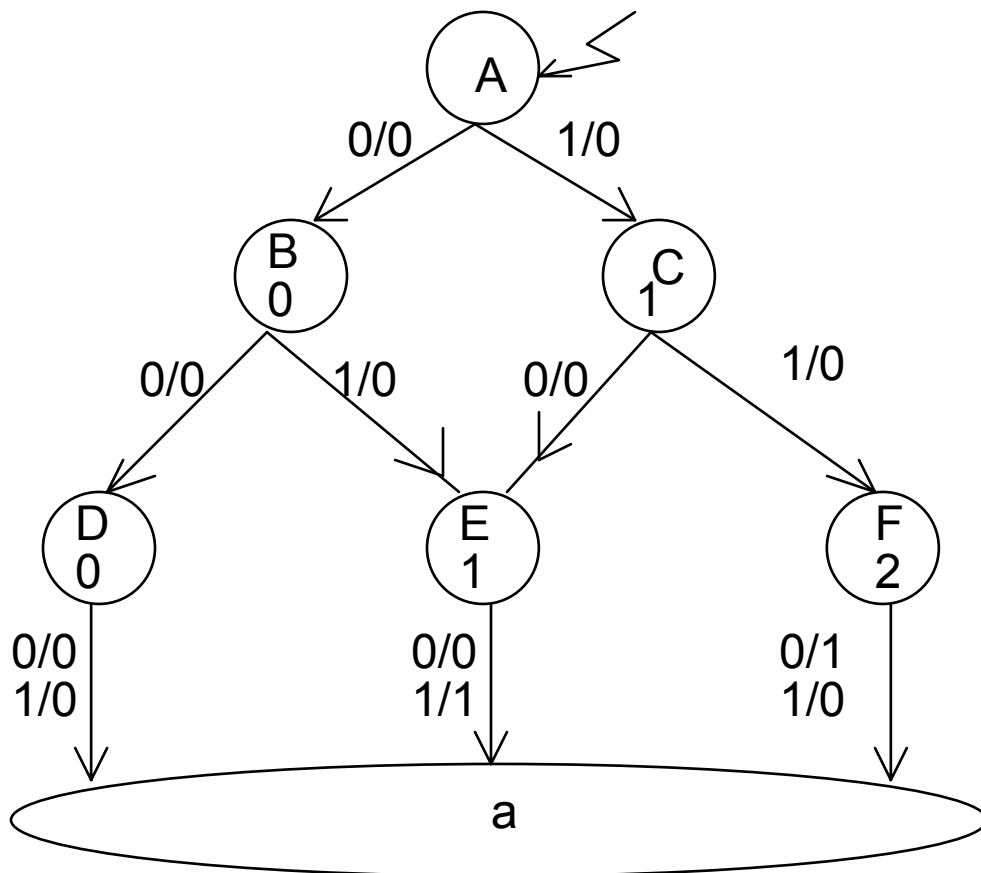
Az *állapotminimalizálás*ra szisztematikus módszerek léteznek (lsd. később). Ezek kiindulópontja az előzetes állapottábla, ez alapján készítik el a minimalizált állapottáblát.

Minimalizált állapotgráf:



Az állapotokba beírtuk, hogy mit jegyeznek meg (eddig hány 1-es jött).

A további eljárások kiindulópontja a *minimalizált állapotábra*.



Az állapotábrát az állapotgráf alapján töltjük ki:

	x=0	x=1
A	B/0	C/0
B	D/0	E/0
C	E/0	F/0
D	A/0	A/0
E	A/0	A/1
F	A/1	A/0

3. Állapotkódolás

Az állapotokhoz kódokat kell rendelni. Erre szintén vannak *szisztematikus módszerek* (lsd. később).

Itt 'ad hoc' módszerrel („hasraütésre”) rendelünk kódot az állapotokhoz. Az állapotot a szekunder változók (Q2Q1Q0) egy kombinációja reprezentálja.

Q2Q1Q0	x=0	x=1
A 000	001/0	010/0
B 001	011/0	100/0
C 010	100/0	101/0
D 011	000/0	000/0
E 100	000/0	000/1
F 101	000/1	000/0

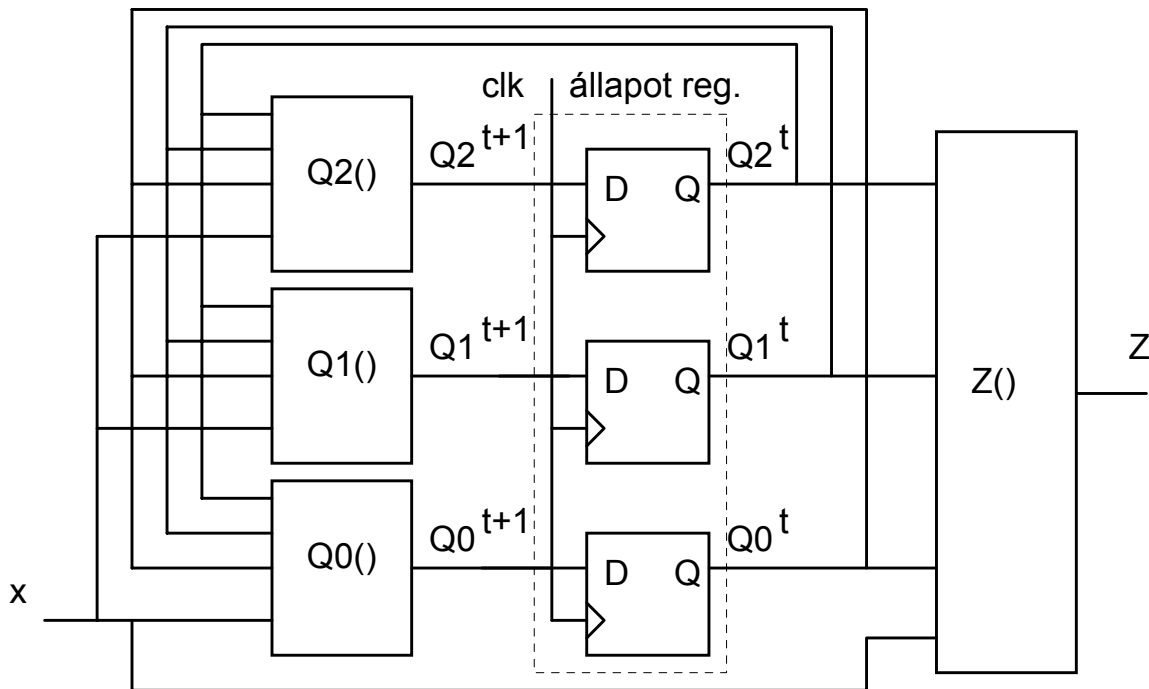
A maradék állapotkódokhoz is ki kell tölteni az állapottáblát.

Q2Q1Q0	x=0	x=1
000	001/0	010/0
001	011/0	100/0
010	100/0	101/0
011	000/0	000/0
100	000/0	000/1
101	000/1	000/0
110	---/-	---/-
111	---/-	---/-

- Ha a többi (normál működésnél nem használt) állapotkódhoz ***közömbös bejegyzést*** teszünk, több egyszerűsítési lehetőség adódhat a kombinációs hálózatrész tervezésénél.
- Ha ***megbízhatóbb működésre*** törekszünk, akkor célszerű előírnunk, hogy ***a normál működésnél nem használt állapotkódokból is a használtakba vezessen a hálózat.*** (Normál működésnél nem használt állapotkódba kerülhet a hálózat külső zavar hatására. Közömbös kitöltésnél szerencsétlen esetben itt ragadhat az automata, ha a következő állapotok kódja szintén nem használt állapotkódba vezet!)

4. Szekunder változók és a kimenet függvényeinek meghatározása

A megtervezendő automata struktúrája:



Megtervezendők a szekunder változók

$$Q2^{t+1}(Q2^t, Q1^t, Q0^t, x),$$

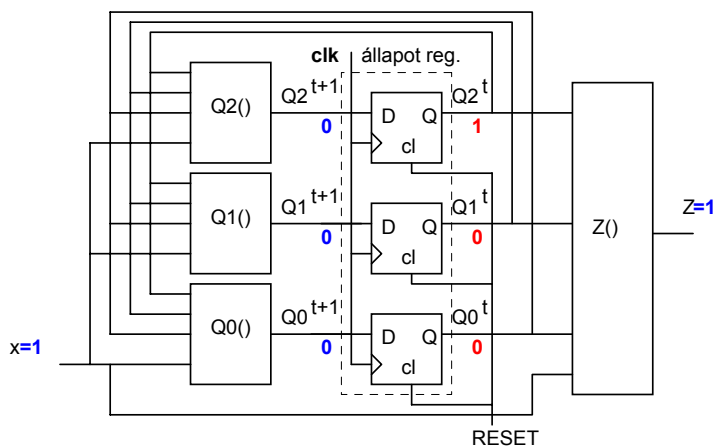
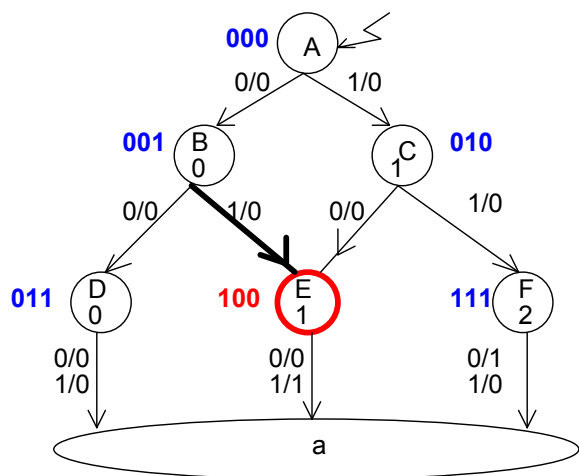
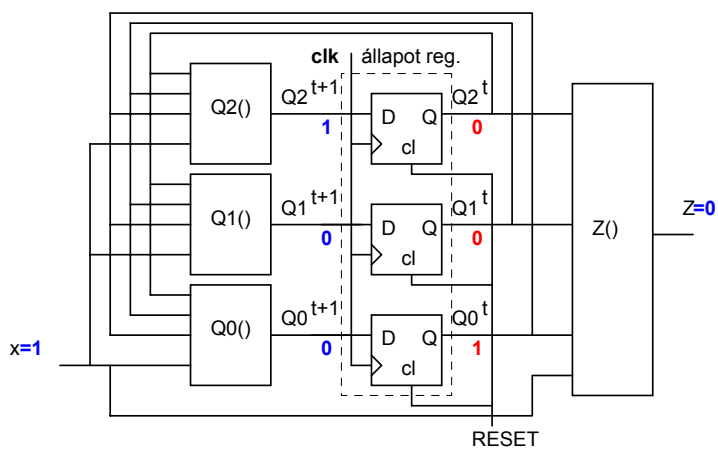
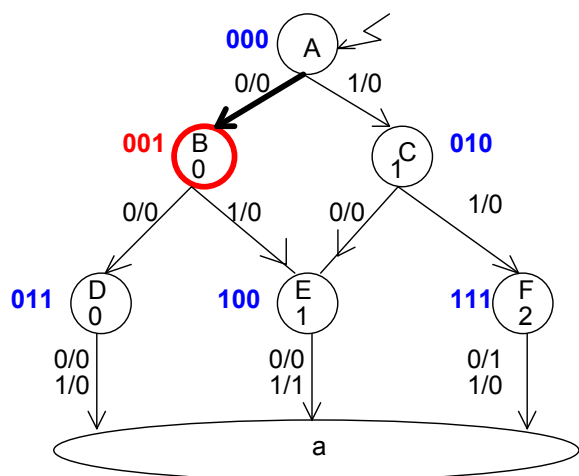
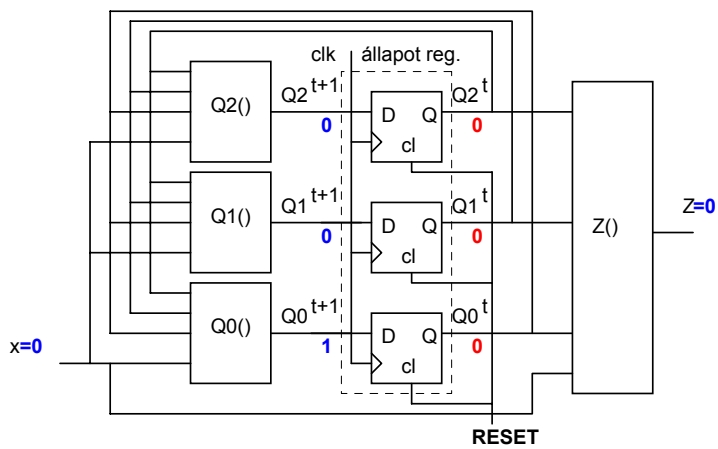
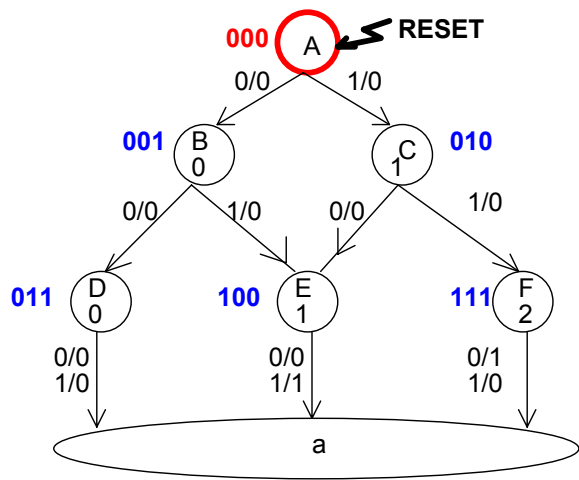
$$Q1^{t+1}(Q2^t, Q1^t, Q0^t, x),$$

$$Q0^{t+1}(Q2^t, Q1^t, Q0^t, x),$$

és a kimenet $Z(Q2^t, Q1^t, Q0^t, x)$ függvényei.

Igazságtáblájukat egyszerre tartalmazza a kódolt állapottábla. Ez alapján lehet az egyes függvények Karnaugh tábláit kitölteni.

Mit csinál az automata RESET után az X=011 bemeneti sorozat hatására?



Q2, Q1, Q0, Z igazságtábláját egyszerre tartalmazza a kódolt állapottábla. Ez alapján lehet az egyes függvények Karnaugh tábláit kitölteni.

Q2Q1Q0	x=0	x=1
A 000	001/0	010/0
B 001	011/0	100/0
C 010	100/0	101/0
D 011	000/0	000/0
E 100	000/0	000/1
F 101	000/1	000/0
110	---/-	---/-
111	---/-	---/-

$Q0^{t+1}$

1			
	1	-	-
		-	-
1			

x

Q1
Q0

$Q1^{t+1}$

	1		
		-	-
		-	-
1			

x

Q1
Q0

$Q2^{t+1}$

1	1	-	-
		-	-
	1		

x

Q1
Q0

Z

		1	
		-	-
		-	-
			1

x

Q1
Q0

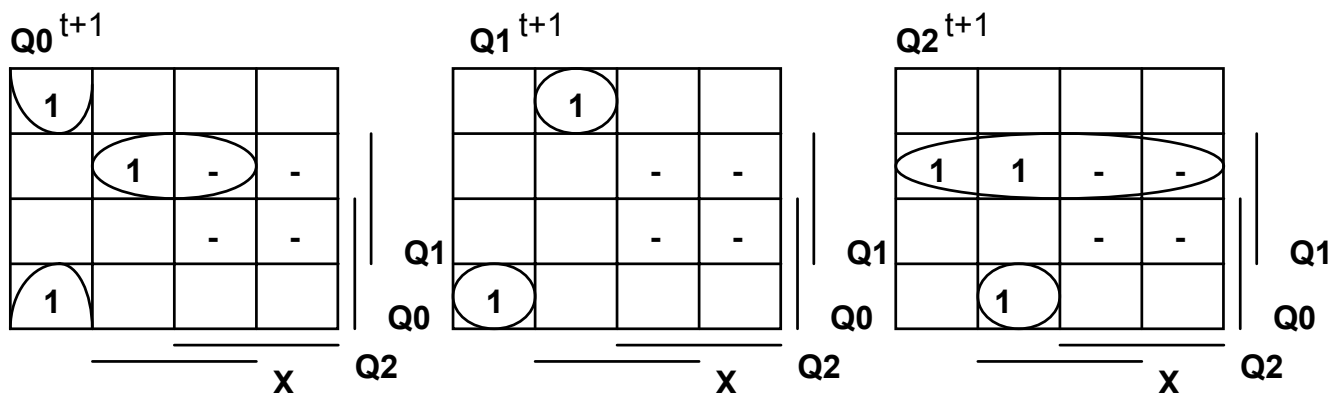
5. Flip-flop választás

Az állapotregisztert különféle flip-flopokkal lehet megvalósítani (D, J-K, T). A flip-flop választás befolyásolja a kombinációs hálózatrész bonyolultságát.

6. Flip-flop vezérlő függvények meghatározása

Szinkron sorrendi hálózat flip-flop vezérlő függvényeit nem kell hazárdmentesíteni! (Két órajel közötti időben lezajlanak a hazárdok.)

Ha D flip-flopot választunk, akkor a flip-flop vezérlő függvénye a megfelelő szekunder változó függvénnyel megegyezik, mivel $Q^{t+1} = D$.



Egyszerűsítés után:

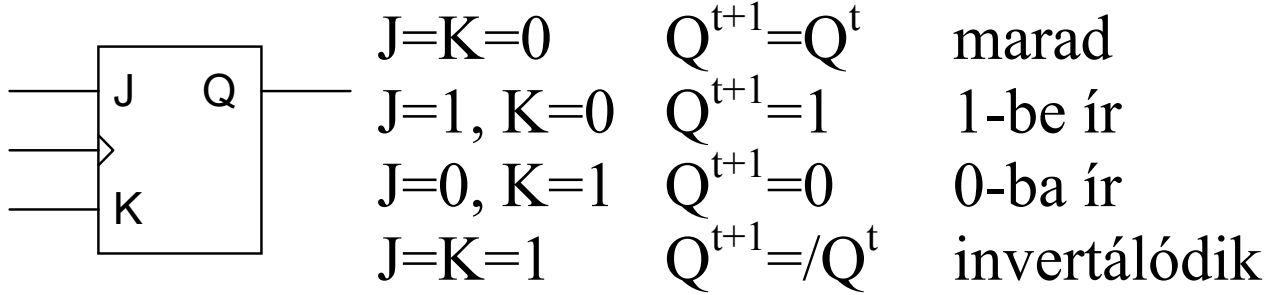
$$Q_0 = x \cdot \overline{Q_0} \cdot \overline{Q_1} + x \cdot \overline{Q_2} \cdot \overline{Q_1} \quad 8 \text{ kapu bemenet}$$

$$Q_1 = \overline{x} \cdot \overline{Q_0} \cdot \overline{Q_1} \cdot \overline{Q_2} + x \cdot \overline{Q_0} \cdot \overline{Q_1} \cdot \overline{Q_2} \quad 10 \text{ bemenet}$$

$$Q_3 = \overline{Q_0} \cdot \overline{Q_1} + x \cdot \overline{Q_0} \cdot \overline{Q_1} \cdot \overline{Q_2} \quad 8 \text{ bemenet}$$

Összesen 26 kapu bemenet

A J-K flip-flop és működése



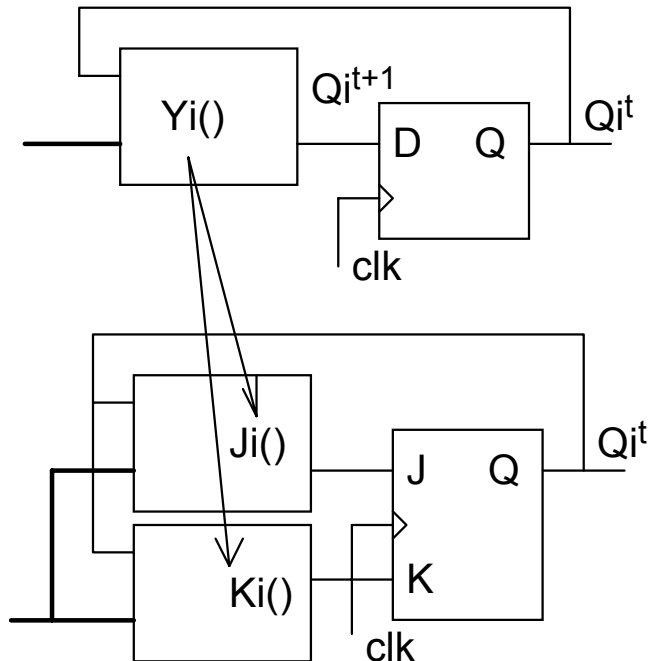
Állapottáblája

Q^{t+1}					
0	0	1	1		
1	0	0	1	Q^t	
		K		J	

Függvénye:

$$Q^{t+1} = J\overline{Q^t} + \overline{K}Q^t$$

Hogyan kell vezérelni a J-K flip-flopot, hogy a szekunder változójának függvénye által előírt legyen a következő állapota?



$Q_i^t Q_i^{t+1}$	J_i	K_i
00	0	-
01	1	-
10	-	1
11	-	0

$Q_i^t Q_i^{t+1}$	J_i	K_i
00	0	-
01	1	-
10	-	1
11	-	0

$$Q^{t+1} = J\overline{Q^t} + \overline{K}Q^t$$

$$Q^t = 1 \rightarrow Q^{t+1} = J \rightarrow J = \overline{Q^{t+1}}$$

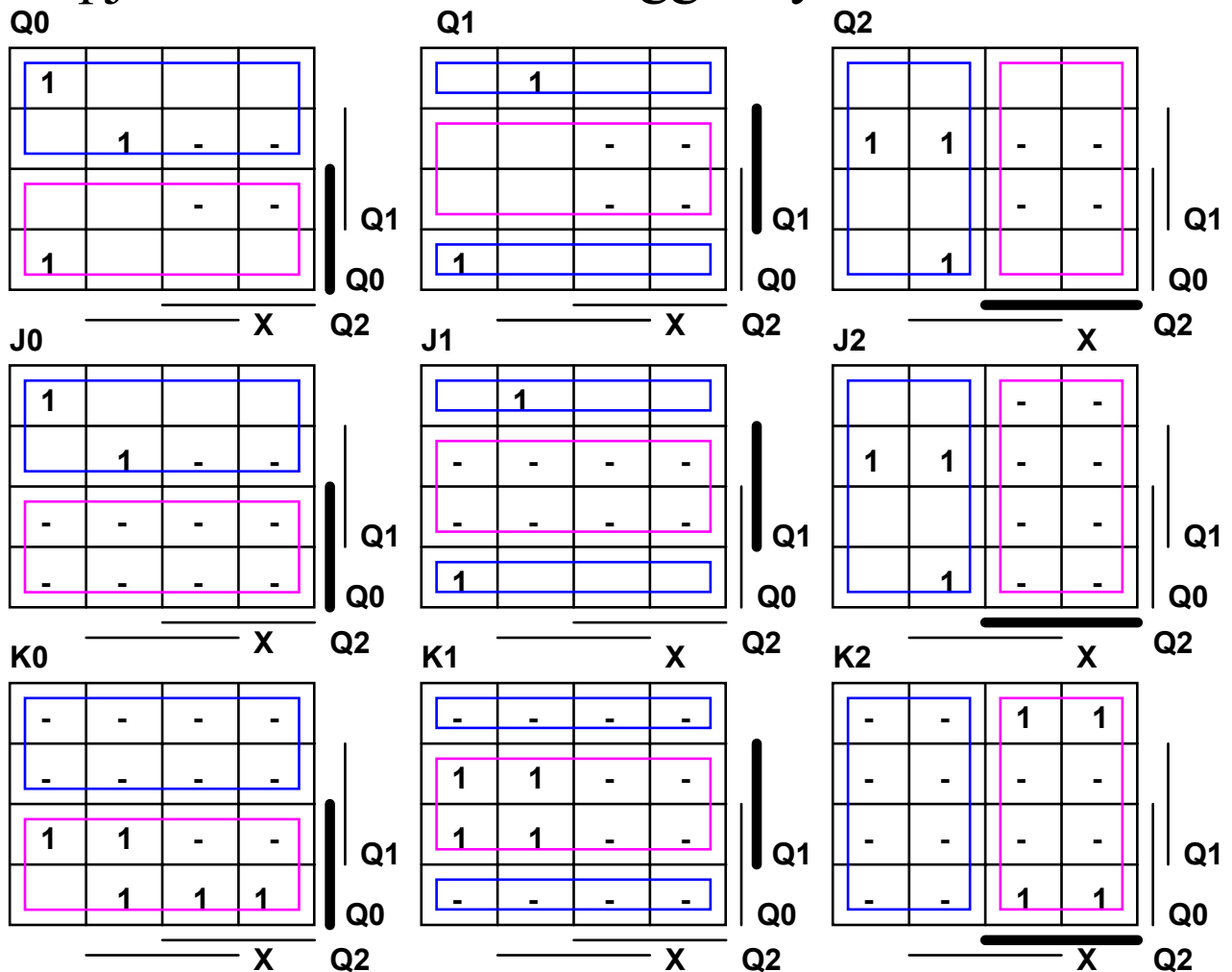
$$Q^t = 0 \rightarrow Q^{t+1} = \overline{K} \rightarrow K = \overline{\overline{Q^{t+1}}}$$

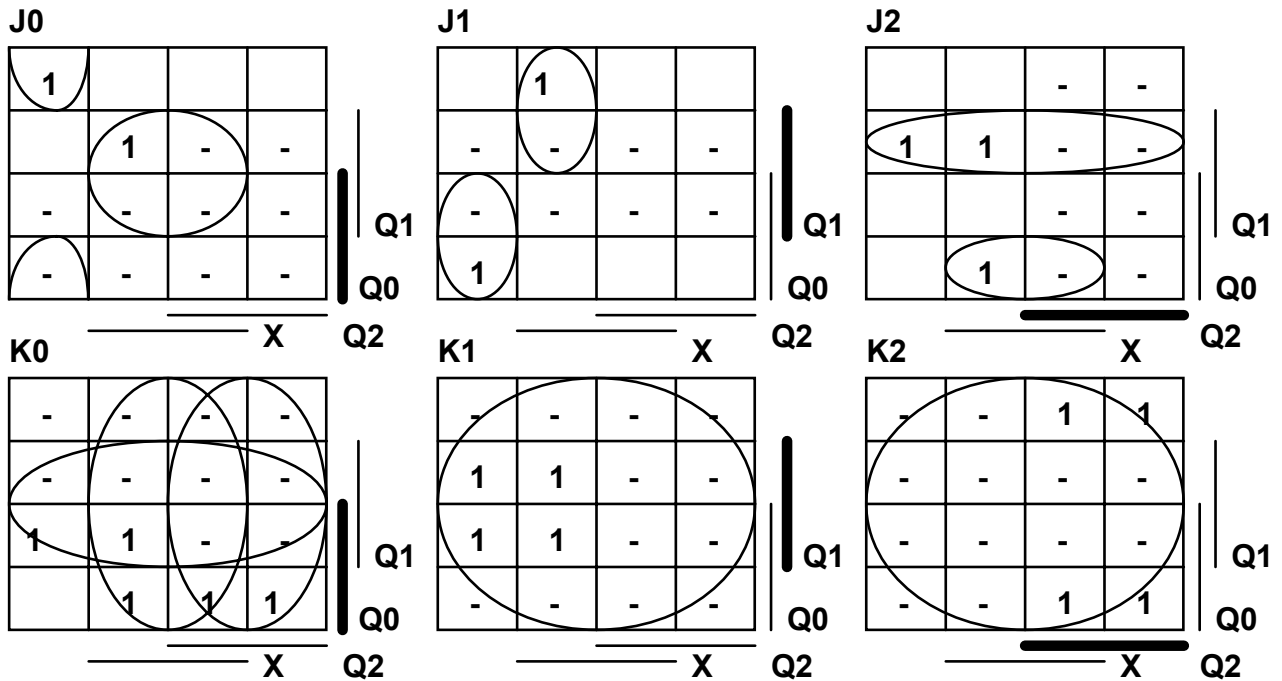
Egyszerű szabály:

$$Q_i^t = 0 \Rightarrow J_i = Q_i^{t+1} \quad \text{és} \quad K = -$$

$$Q_i^t = 1 \Rightarrow J_i = - \quad \text{és} \quad K = /Q_i^{t+1}$$

A feladat szekunder változóinak függvényei alapján a J és K vezérlőfüggvények:





$$J0 = X \cdot Q1 + \neg X \cdot \neg Q1 \cdot \neg Q2$$

$$K0 = Q1 + Q2 + X$$

$$J1 = X \cdot \neg Q0 \cdot \neg Q2 + \neg X \cdot Q0 \cdot \neg Q2$$

$$K1 = 1$$

$$J2 = \neg Q0 \cdot Q1 + X \cdot Q0 \cdot \neg Q1$$

$$K2 = 1$$

7 kapu bemenet

3 bemenet

8 bemenet

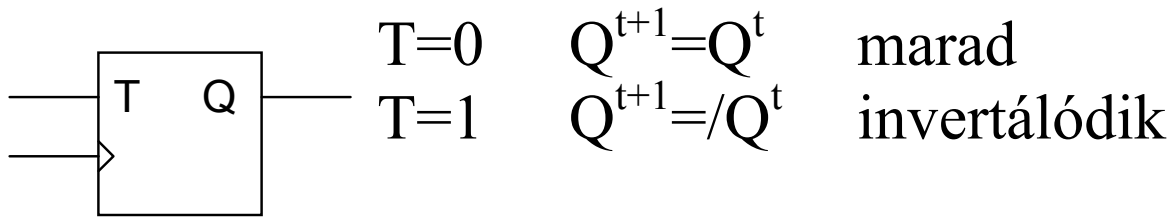
0 bemenet

7 bemenet

0 bemenet

Összesen 25 kapu bemenet. (D-vel 26 volt.)

A T flip-flop és működése

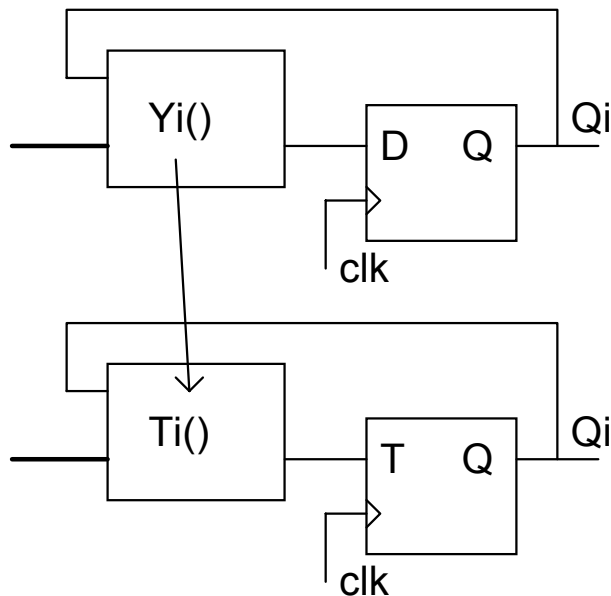


Állapottáblája

Q^{t+1}		
0	1	Q^t
1	0	
		T

Függvénye: $Q^{t+1} = T\overline{Q^t} + \overline{T}Q^t$

Hogyan kell vezérelni a T flip-flopot, hogy a szekunder változójának függvénye által előírt legyen a következő állapota?

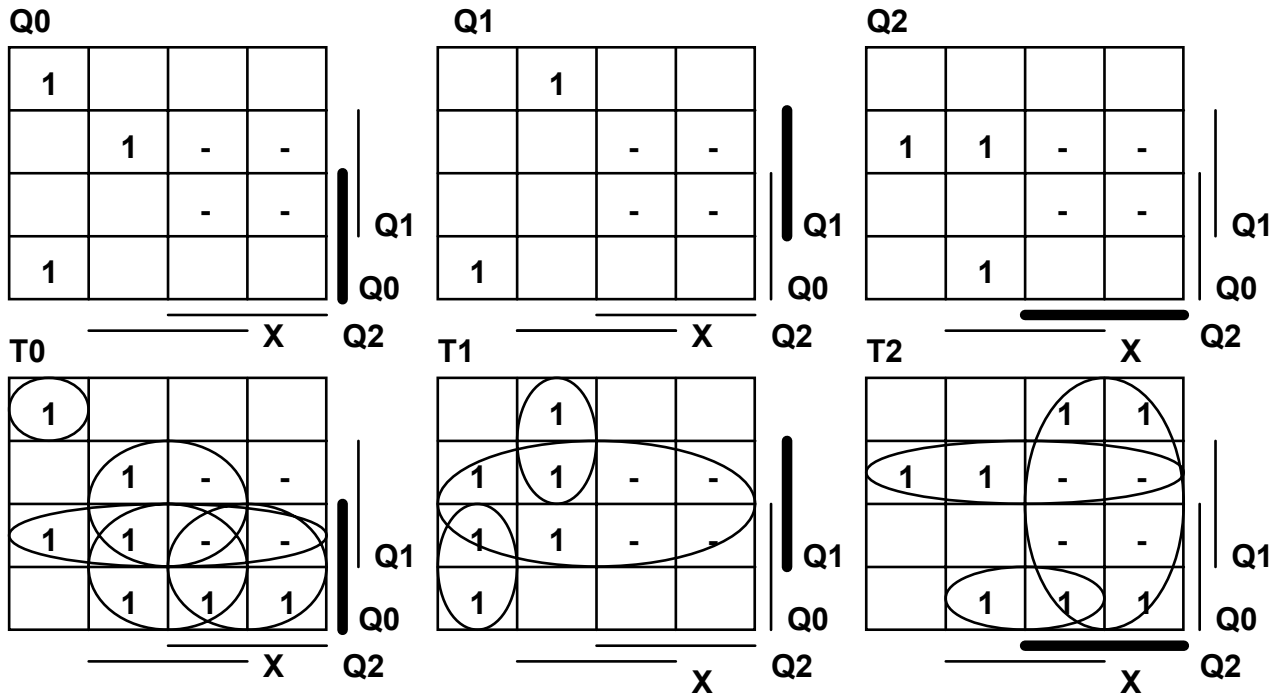


$Q_i^t Q_i^{t+1}$	T_i
00	0
01	1
10	1
11	0

$$Q^{t+1} = T\overline{Q^t} + \overline{T}Q^t$$

$$Q^t = 0: Q^{t+1} = T \rightarrow T = Q^{t+1} \quad Q^t = 1: Q^{t+1} = \overline{T} \rightarrow T = \overline{Q^{t+1}}$$

Egyszerű szabály: $Q_i^t=0 \Rightarrow T_i= Q_i^{t+1}$
 $Q_i^t=1 \Rightarrow T_i= /Q_i^{t+1}$



$$T0 = \overline{X} \cdot \overline{Q0} \cdot \overline{Q1} \cdot \overline{Q2} + X \cdot Q1 + Q0 \cdot Q1 + X \cdot Q0 \quad 14$$

$$T1 = Q1 + X \cdot \overline{Q0} \cdot \overline{Q2} + \overline{X} \cdot Q0 \cdot \overline{Q2} \quad 10$$

$$T2 = \overline{Q0} \cdot Q1 + Q2 + \overline{X} \cdot Q0 \cdot \overline{Q1} \quad 8$$

Összesen 32 kapu bemenet.

7. Kapcsolási rajz

A J-K flip-flopos megvalósítás a legolcsóbb.

$$J_0 = x \cdot Q_1 + \overline{x} \cdot \overline{Q_1} \cdot \overline{Q_2}$$

$$K_0 = Q_1 + Q_2 + X$$

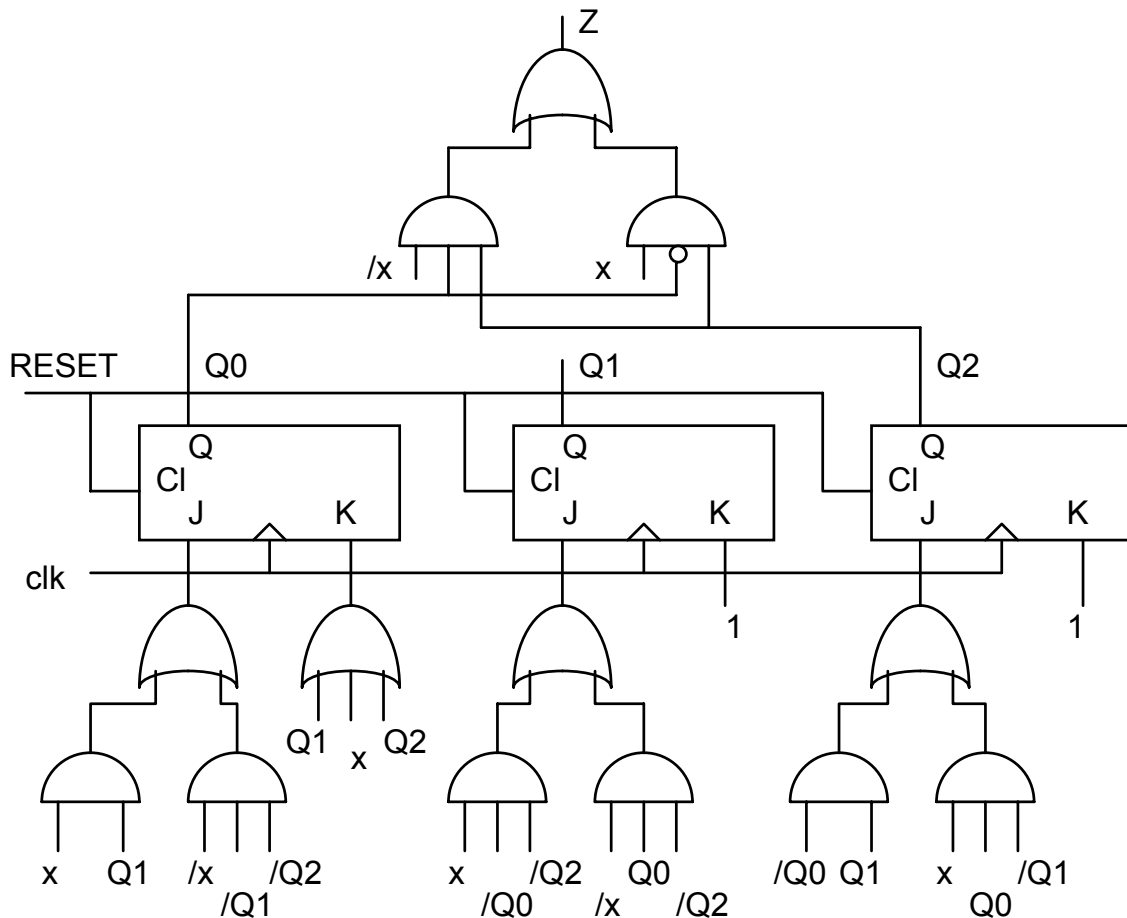
$$J_1 = X \cdot \overline{Q_0} \cdot \overline{Q_2} + \overline{x} \cdot Q_0 \cdot \overline{Q_2}$$

$$K_1 = 1$$

$$J_2 = \overline{Q_0} \cdot Q_1 + X \cdot Q_0 \cdot \overline{Q_1}$$

$$K_2 = 1$$

$$Z = \overline{x} \cdot Q_0 \cdot Q_2 + X \cdot \overline{Q_0} \cdot Q_2$$



Flip-flopok egymásba alakítása

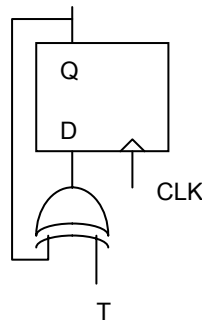
A különféle flip-flopok megvalósíthatók egymással. Amely flip-flopot meg akarjuk valósítani, annak az állapottáblája lesz a kiinduló kódolt állapottábla. Amely flip-flopot felhasználjuk a megvalósításra, annak a vezérlő függvényeit kell a megvalósítandó flip-flop állapottáblája alapján megterveznünk.

D-ből T

Ha D flip-flopot használunk a megvalósításra, akkor egyszerűen a megvalósítandó flip-flop állapottáblája a vezérlőfüggvény.

Q^{t+1}			
0	1		
1	0		
		Q^t	\overline{T}

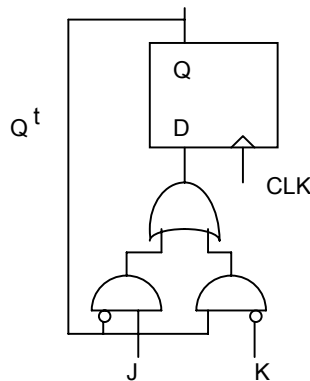
0	1		
1	0		
		Q^t	\overline{T}



$$D = T\overline{Q}^t + \overline{T}Q^t$$

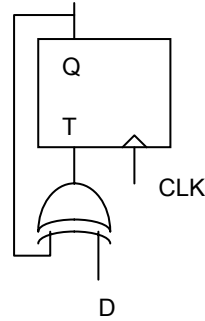
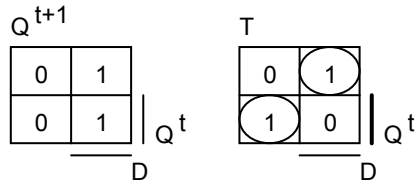
D-ből JK

Q^{t+1}				
		1	1	
1				1
				Q^t
		\overline{K}	J	



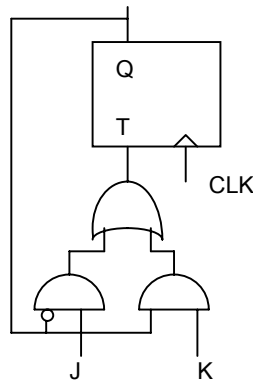
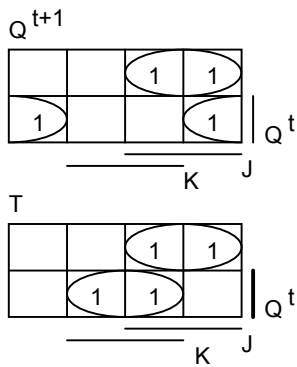
$$D = J\overline{Q}^t + \overline{K}Q^t$$

T-bóil D



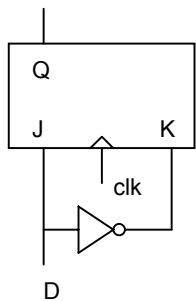
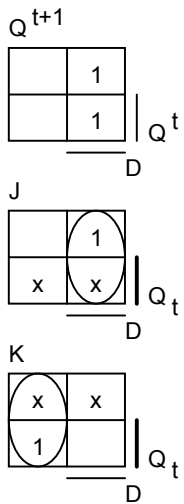
$$T = D\overline{Q}^t + \overline{D}Q^t$$

T-bóil JK



$$T = J\overline{Q}^t + KQ^t$$

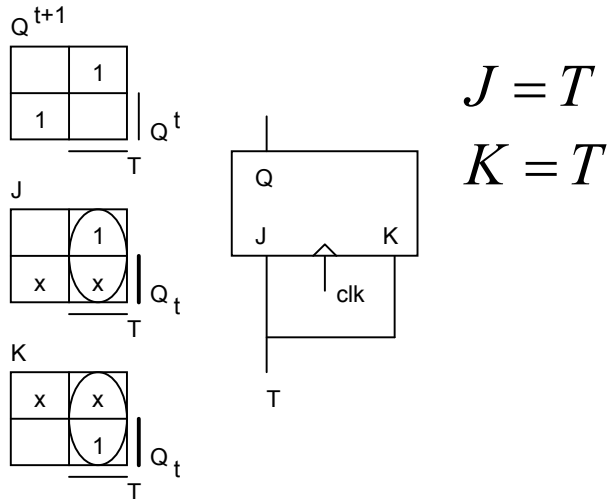
JK-bóil D



$$J = D$$

$$K = \overline{D}$$

JK-ból T



Gyakorló feladat

Tervezzen konfigurálható flip-flopot, JK-ból.

A speciális flip-flop bemenetei:

V2, V1 konfiguráló bemenetek TDJ, K, a flip-flop bemenetek.

Viselkedése:

V2	V1	
0	-	JK flip-flop: $J=TDJ$, $K=K$
1	0	D flip-flop: $D=TDJ$
1	1	T flip-flop: $T=TDJ$

A megvalósításhoz ÉS, VAGY, EXOR kapukat használhat. Törekedjen az egyszerűsége!