

BUDAPESTI MŰSZAKI EGYETEM
VILLAMOSMÉRNÖKI KAR, SZAKMÉRNÖKI TAGOZAT
MÉRNÖKI TOVÁBBKÉPZŐ INTÉZET KIADVÁNYA

V. 113.

Bohus Miklós

DIGITÁLIS RENDSZEREK TERVEZÉSE

KÉZIRAT

TANKÖNYVKIADÓ, BUDAPEST 1969

75,- Ft

BUDAPESTI MŰSZAKI EGYETEM
VILLAMOSMÉRNÖKI KAR, SZAKMÉRNÖKI TAGOZAT
MÉRNÖKI TOVÁBBKÉPZŐ INTÉZET KIADVÁNYA

V. 113.

Bohus Miklós

DIGITÁLIS RENDSZEREK TERVEZÉSE

KÉZIRAT

TANKÖNYVKIADÓ, BUDAPEST 1969

Lektor:

Dr. Gertler János
műszaki tudományok kandidátusa

A kiadásért felelős:

a Budapesti Műszaki Egyetem Villamosmérnöki Karának dékánja
Megjelent a Tankönyvkiadó Vállalat műszaki gondozásában

Műszaki vezető: Hátori József

Műszaki szerkesztő: Villányi Zsuzsa

Megrendelve: 1969. március. Megjelent: 1969. július. Példányszám: 253

Készült: Rotaprint lemezről (kicsinyítéssel), a MSZ 5601-59

és MSZ 5602-55 szabványok szerint,
15,4 (A/5) iv terjedelemben, 128 ábrával

BEVEZETÉS

A különböző digitális berendezések, elsősorban az elektronikus digitális programvezérelt számítógépek megjelenése nagymértékben meggyorsította körünk számos tudományos és technikai problémájának megoldását. A számítógépek felhasználásával olyan hatalmas mennyiségű számítást lehet elvégezni, ami kézi erővel időhiány miatt lehetetlen volna. A gépek elterjedésével kialakult korszerű számítástechnikai eszközök alkalmazása ma még beláthatatlan lehetőségeket nyit meg a modern technika előtt. A nagysebességű számítógépek több millió műveletet végezhetnek el másodpercenként husz-harmincjegyű számokkal, így egy másodperc alatt több műveletet hajtanak végre, mint egy jól begyakorolt számoló hónapok, évek alatt. A digitális berendezések segítségével nemcsak számítási, hanem logikai feladatokat is meg tudunk oldani. A gépek alkalmazása lehetőséget ad arra, hogy a monoton, rutinszerű szellemi tevékenységeket számítógépekkel végezzük el. Különösen nagy jelentősége van a digitális berendezések felhasználásának ipari folyamatok automatikus vezérlésében és szabályozásában.

A "Digitális rendszerek tervezése I." c. jegyzet a digitális technika alapfogalmaival, szekvenciális logikai hálózatok felépítésével, áramköri megvalósításával, szinkron működésű szekvenciális hálózatok tervezésének bevezető jellegű ismereteivel foglalkozik.

A jegyzet a "Félvezetős digitális elektronika" szakmérnöki tanfolyam első éves hallgatói számára készült. A korlátozott terjedelemből való tekintettel a szekvenciális hálózatok, funkcionális egységek optimális tervezésének kérdéseit részletesen nem tárgyalja. Az aszinkron működésű szekvenciális rendszerek problémáit a jegyzet következő kötete tartalmazza. 2

Az alapfogalmak, működési elvek, tervezési módszerek megértésén kívül az anyag önálló alkalmazására, a tervezési készség kialakítására különös gondot fordítottunk. Ezt szolgálja a viszonylag bő példaanyag, valamint a nagyszámú gyakorló feladat. A logikai, rendszertechnikai, aritmetikai, vezérlési feladatok elsősorban a digitális számítógépek tervezéséhez nyújtanak bevezető jellegű ismereteket.

Az anyag tárgyalásában a jegyzet több helyen a szerzőnek e témában az irodalomban nem szereplő önálló eredményeit is tartalmazza.

Az elnevezések, jelölések megegyeznek a Budapesti Műszaki Egyetem Villamosmérnöki Kar Híradástechnikai szakán az alapozó szaktárgyakban használt elnevezésekkel és jelölésekkel.

Ezúton mondok köszönetet dr. Gertler János kandidátusnak, a kézirat lektorának lelkiismeretes, gondos munkájáért, aki hasznos észrevételeivel, értékes tanácsaival a jegyzet végleges kialakításában nagy segítséget nyújtott.

1. SZEKVENCIÁLIS HÁLÓZATOK

1.1 Bevezetés

A logikai hálózatok, mint ismeretes lehetnek

- a) kombinációs vagy kapcsoló hálózatok
- b) szekvenciális vagy sorrendi hálózatok.

A szekvenciális hálózatok egyéb elnevezése időrendi vagy többütemű hálózatok.

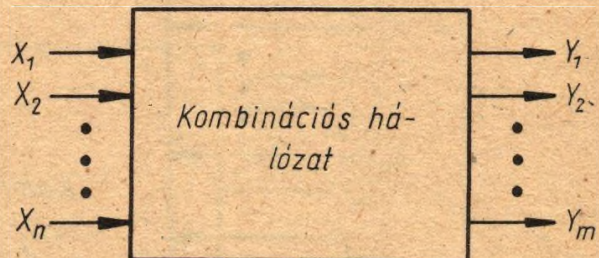
Kombinációs hálózat kimenő változóinak ($Y_1, Y_2, \dots, Y_m$) értékét valamely időpontban a bemenő változók ($X_1, X_2, \dots, X_n$) ugyanazon időpontbeli értékei egyértelműen meghatározzák (1-1. ábra). A kimenő változók tehát a bemeneti változók logikai függvényeivel állíthatók elő:

$$Y_1 = F_1(X_1, X_2, \dots, X_n)$$

$$Y_2 = F_2(X_1, X_2, \dots, X_n)$$

$\vdots$

$$Y_m = F_m(X_1, X_2, \dots, X_n)$$



1-1. ábra

Kombinációs hálózat jelképi jelölése

$$Y_i = F_i(X_1, X_2, \dots, X_n)$$

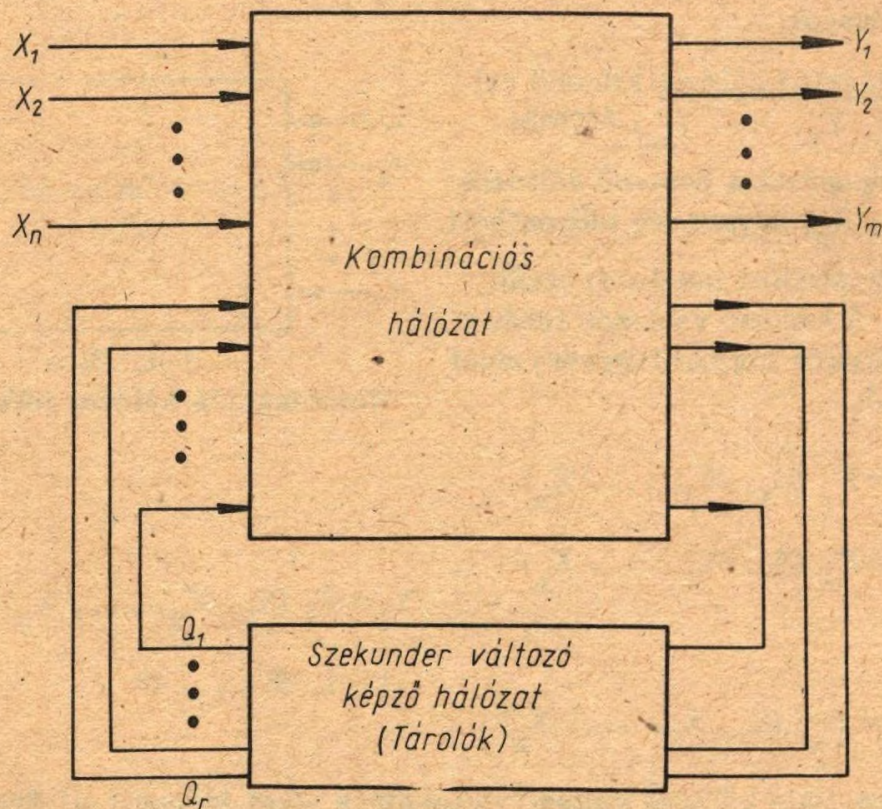
$$i = 1, 2, \dots, m$$

Szekvenciális hálózat kimenő változóinak adott időpontbeli értéke nemcsak a bemenő változók ugyanazon időpontbeli értékeitől függ, hanem a bemenő változók vezérlésének sorrendjétől, a hálózat előző vezérlési állapotától is. A bemenő változók azonos kombinációihoz a kimenő változók más és más értéke tartozhat attól függően, hogy milyen időpontban lép fel az adott bemeneti kombináció. Mivel a bemeneti változók kombinációján kívül ezek megjelenési ideje (sorrendje) együttesen határozza meg a kimenő változók értékét, ezért az ilyen tulajdonságú logikai hálózatokat szekvenciális (vagy sorrendi) hálózatoknak nevezik.

A szekvenciális hálózatokban tehát egy új változó is fellép, az idő, mert a bemenő változók adott kombinációjához más-más időpontban különböző kimenő jelet rendelhetünk.

A bemeneti változók nem határozzák meg egyértelműen a kimeneti változók értékét, ezért a bemeneti változókat újabb belső változókkal kell kiegészítenünk. A belső (vagy szekunder) változók egyben rögzítik (tárolják) a hálózat előző vezérlési állapotait, és a bemenő változókkal együtt már egyértelműen meghatározzák a kimenő változók értékét. A belső változókat tároló elemek (szekunder elemek) segítségével valósítják meg, melyek vezérlését a bemenő változók végzik. A belső változókat realizáló tárolók állapotai a hálózat belső állapotai ($Q_1, Q_2, \dots, Q_r$). A belső állapotok a bemenő változók és a tárolók állapotaitól függenek.

A szekvenciális hálózatok tömbvázlata (1-2. ábra) tehát tartalmaz a kimenő változók előállítására szolgáló kombinációs hálózatot és a belső állapotok (szekunder változók) előállítására szolgáló szekunder változó képző (tároló) hálózatot.



1-2. ábra
Szekvenciális hálózat tömbvázlata

A szekvenciális hálózatokat két csoportra oszthatjuk:

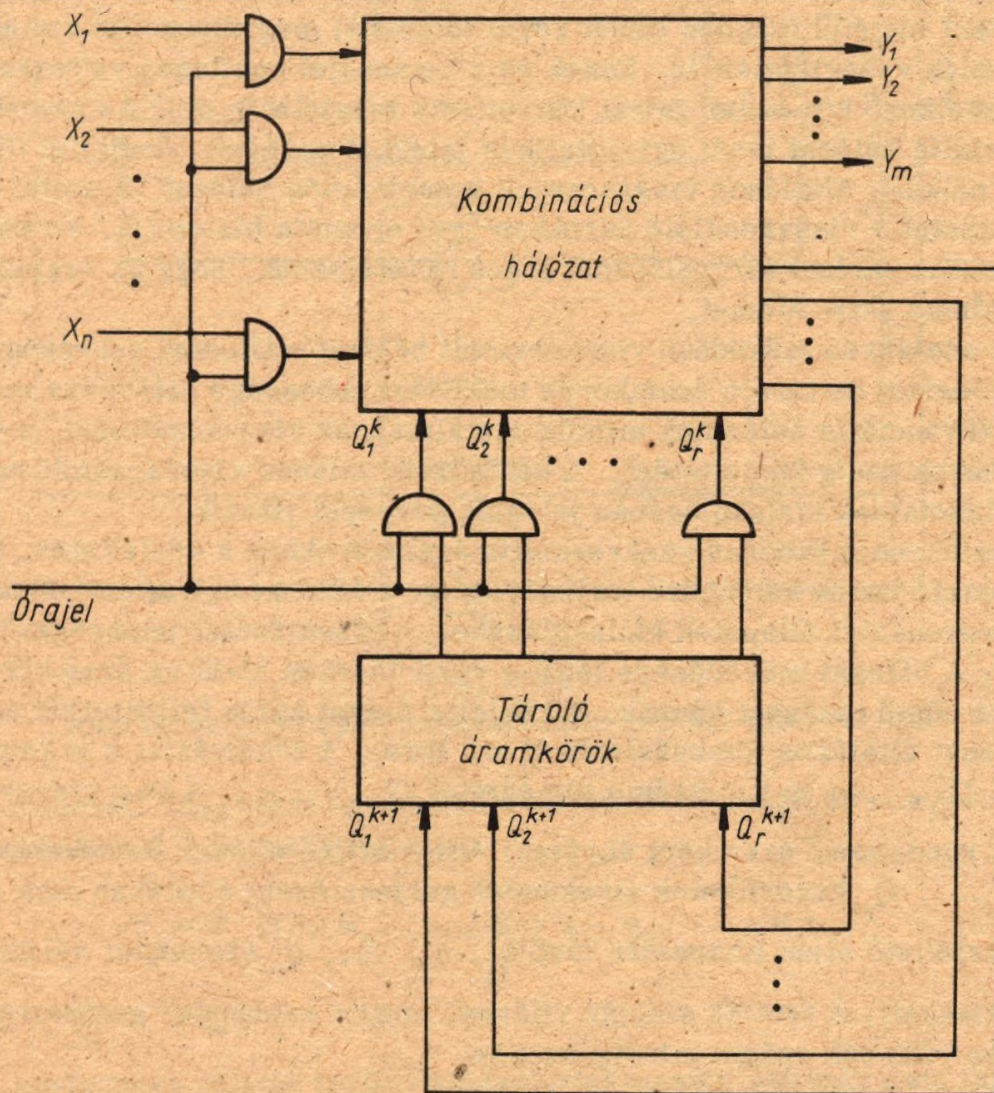
- Szinkron, órajellel működő vagy impulzus üzemi szekvenciális hálózatok.
- Aszinkron, vagy órajel nélkül működő szekvenciális hálózatok.

1.2 Szinkron szekvenciális hálózatok

Szinkron szekvenciális hálózatok kombinációs hálózatának valamennyi bemenőjelét általában órajellel (szinkronizáló jellel) kapuzzuk. A szinkron működésű szekvenciális hálózat a bemenőjelnek az órajel pillanatában fellépő értékeit hasznosítja. A szekunder változókat tartalmazó belső állapot tárolókat ugyancsak az órajel beérkezésének pillanatában olvassuk ki. A visszacsatolást tartalmazó szekvenciális áramkör tárolóinak állapotát a $(k+1)$ -ik ütemben a bemenő változók és a tárolók k -adik ütemben felvett értékei határozzák meg (1-3. ábra):

$$Q_i^{k+1} = G_i(X_1^k, X_2^k, \dots, X_n^k; Q_1^k, Q_2^k, \dots, Q_r^k)$$

ahol $i = 1, 2, \dots, r$.



1-3. ábra

Szinkron működésű szekvenciális hálózat tömbvázlata

A kimeneti változók k -adik ütembeli értékét a bementi változók és a szekunder változók (belső állapotok) k -adik ütemben felvett értéke határozza meg:

$$Y_i^k = F_i(X_1^k, X_2^k, \dots, X_n^k; Q_1^k, Q_2^k, \dots, Q_r^k)$$

ahol $i = 1, 2, \dots, m$.

1.3 Aszinkron szekvenciális hálózatok

Aszinkron működésű szekvenciális hálózat kombinációs hálózatának bemenőjeleit nem kapuzzuk. A bemenőjelek és a szekunder változók jelének változását közvetlenül érzékeli a kombinációs hálózat, és a felhasznált áramkörök késleltetési idejétől függően (tehát rövid idő alatt) megváltoztatja kimenőjeleit, így a tárolók vezérlőjeleit is. Ennek és a bemenőjel esetleges változásának hatására keletkező kimenőjel ismét közvetlenül megjelenik stb. Az aszinkron szekvenciális hálózat tehát egy sztatikus jelekkel vezérelt, órajellel történő kapuzás nélküli, általános visszacsatolt kombinációs hálózat. A stabil egyensúlyi állapotot a visszacsatoló hurokban levő erősítés biztosítja. Az aszinkron szekvenciális hálózat tárgyalásánál eleve feltételezzük, hogy az áramkör mindig tartalmaz aktiv elemet.

Az aszinkron működésű visszacsatolt hálózat működési sebességét a felhasznált logikai és tároló áramkörök működési sebessége határozza meg.

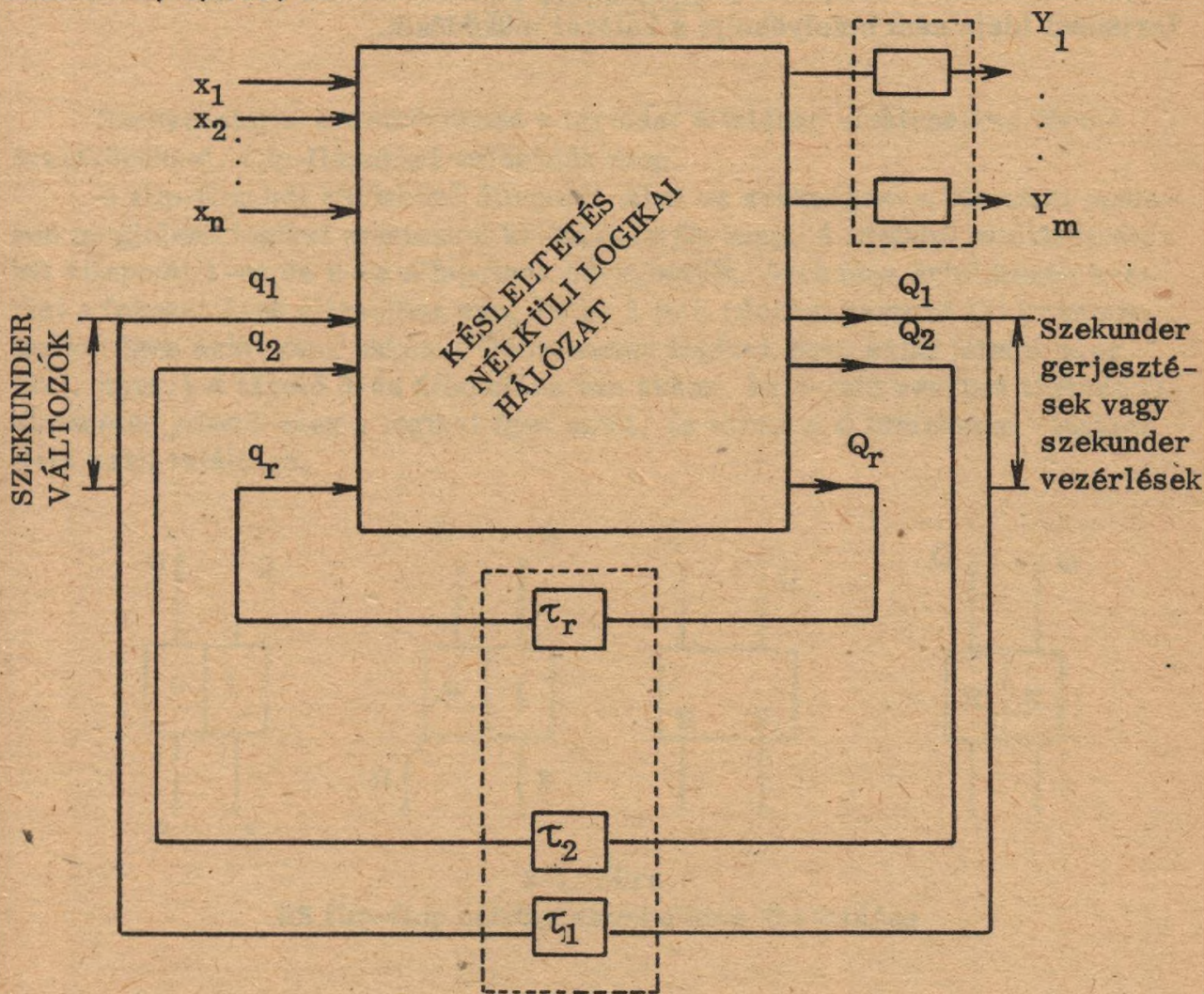
Szekvenciális hálózatok mindig tartalmaznak visszacsatolást, a visszacsatoló hurok pedig késleltetést. A késleltetés minden visszacsatolt rendszer stabil állapotainak kialakulásában jelentős szerepet játszik.

Aszinkron működésű szekvenciális áramkörökben a késleltetés, több visszacsatoló hurok különböző terjedési ideje a hálózat helytelen működéséhez, instabil egyensúlyi állapotok kialakulásához, versenyfutási jelenségekhez stb. vezethet. A hálózat működésére tehát a vezérléseken kívül az áramköri késleltetések is döntő hatással vannak. A késleltetéseket külön feltüntetjük az aszinkron hálózat általános tömbvázlatán (1-4. ábra). A tömbvázlat n számú bemenettel ($X_1, \dots, X_n$) m számú kimenettel ($Y_1, \dots, Y_m$) és r visszacsatoló hurokkal rendelkező áramkört ábrázol. A késleltető elemek bemenetén levő $Q_1, Q_2, \dots, Q_r$ vezérlőjelek (szekunder gerjesztések) általában nem egyeznek meg a késleltető elem kimenetén levő $q_1, q_2, \dots, q_r$ szekunder változókkal (válaszjelekkel). A fentiek alapján világos, hogy a szekunder gerjesztések és szekunder változók száma mindig azonos.

A szekunder gerjesztéseket a bemeneti változók és a szekunder változók függvényeivel állíthatjuk elő:

$$Q_i = G_i (X_1, X_2, \dots, X_n; q_1, q_2, \dots, q_r)$$

ahol $i = 1, 2, \dots, r$.



1-4. ábra

Aszinkron működésű szekvenciális hálózat tömbvázlata

A kimeneti változókat ugyancsak a bemeneti változók és a szekunder változók segítségével határozhatjuk meg:

$$Y_i = F_i (X_1, X_2, \dots, X_n; q_1, q_2, \dots, q_r)$$

ahol $i = 1, 2, \dots, m$.

A késleltetés nélküli logikai hálózat visszacsatolást már nem tartalmaz. A visszacsatoló hurkok a hálózat belső állapotait, a tárolást valósítják meg. A késleltetés nélküli logikai hálózat tehát mindig kombinációs hálózat.

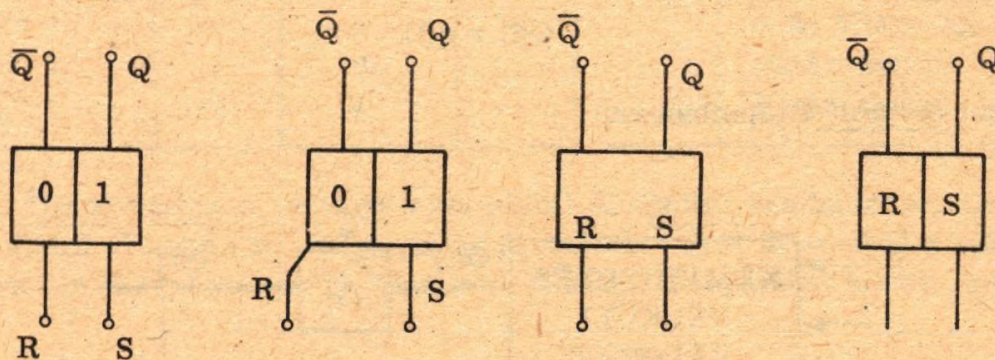
Szinkron működésű szekvenciális áramkörökben a késleltetéseket azért nem tüntettük fel külön, mert a vezérlések hatására létrejövő állapotváltozásokat mindig a következő szinkronizáló impulzus beérkezésekor használjuk fel. Amennyiben biztosítjuk, hogy a visszacsatoló hurkok állapotváltozásai két egymás utáni szinkronizáló (óra) jel időtartama alatt lejátszódjanak, vagyis, hogy a hálózat stabil állapotba jusson, akkor a visszacsatoló hurkok különböző terjedési ideje nem befolyásolja a hálózat működését.

2. ELEKTRONIKUS TÁROLÓ ÁRAMKÖRÖK

2.1 Bevezetés

Szekvenciális áramkörökben a tárolást általában elektronikus tároló áramkörökkel, flip-flopokkal valósítják meg.

A flip-flop két különböző állapotát (ahol ez szükséges) a kimeneti pontokon megjelenő logikai szintekkel különböztetjük meg. A bistabil multivibrátor két állapotát 1-es és 0-ás állapotnak nevezhetjük. Szokásos értelmezés szerint a fokozat 1-es állapotban van, ha az 1 jelű részhez tartozó Q kimeneten logikai Igen szint van; ekkor a $\bar{Q}$ kimeneten logikai Nem szint jelenik meg. (2-1. ábra.) A tároló 0-ás állapotban van akkor, ha 0-jelű részhez tartozó $\bar{Q}$ kimeneten jelenik meg a logikai Igen szint, és ekkor a Q kimeneten logikai Nem szint található.



2-1. ábra

RS flip-flop néhány szimbolikus ábrázolása

2.2 Tároló áramkörök felosztása

Tároló áramkörök közül ebben a fejezetben a következő típusu flip-flopokkal foglalkozunk

- a) RS ill. inverz RS flip-flop
- b) JK ill. inverz JK flip-flop
- c) T ill. inverz T flip-flop
- d) RST flip-flop
- e) D flip-flop

2.2.1. RS flip-flop

Az RS flip-flop SET bemenetét gerjesztve a tároló 1-es, RESET bemenetét gerjesztve 0-ás állapotba kerül. Az R és S bemenetek egyidejű gerjesztésekor az áramkör működése bizonytalanná válik, ezt a vezérlési állapotot a vezérlőkör megfelelő kialakításával ki kell küszöbölnünk. ($R = 1, S = 1$ meg nem engedett kombináció.)

Az RS tároló vezérelt állapotát jelöljük $Q(t_0)$ -al, a hozzátartozó kimenő jelet $q(t)$ -vel. A kimenőjel értéke stabil állapotban megegyezik $Q(t_0)$ gerjesztési függvény értékkel:

$$q(t_0) = Q(t_0)$$

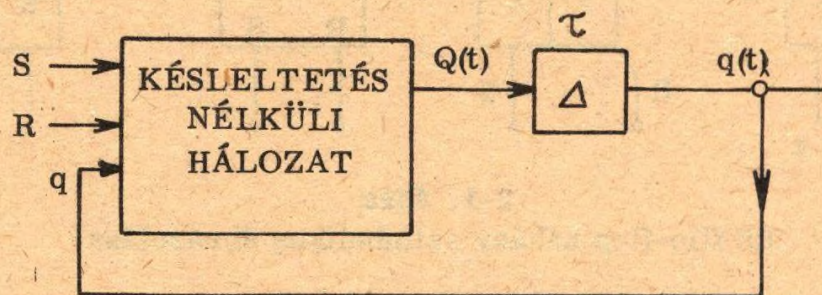
Az egyenletből megállapítható, hogy az RS tároló realizálható aktiv elemet tartalmazó visszacsatolt kombinációs hálózat és késleltetés segítségével (2-2. ábra). A tároló stabil egyensúlyi állapotában (bármely vezérlés utáni egyensúlyi helyzetben):

$$Q(t) = q(t)$$

Amennyiben

$$Q(t) \neq q(t)$$

a hálózat instabil állapotban van.



2-2. ábra

RS flip-flop, mint aszinkron működésű szekvenciális hálózat

Egyelőre az RS tároló lehetséges stabil egyensúlyi állapotait vizsgáljuk, tekintsük a kimenetnek $Q(t)$ jelet.

A tároló működésének leírása több ekvivalens módon történhet.

a) Igazságtáblázat. Irjuk fel az RS flip-flop igazság táblázatát. Az áramkör kimenőjelenek logikai értékét a bemenőjelek valamilyen t_0 kezdőpontban felvett értéke: $R(t_0)$; $S(t_0)$ és az áramkör t_0 időpontbeli állapota $Q(t_0)$ határozza meg. A t_0 időpontbeli vezérlés hatására a kimenőjel ill. a fokozat állapotát jelöljük $Q(\tau) = Q(t_0 + \Delta t)$ -vel. Az 1-es, 2-es táblázaton láthatjuk az RS flip-flop igazság táblázatát.

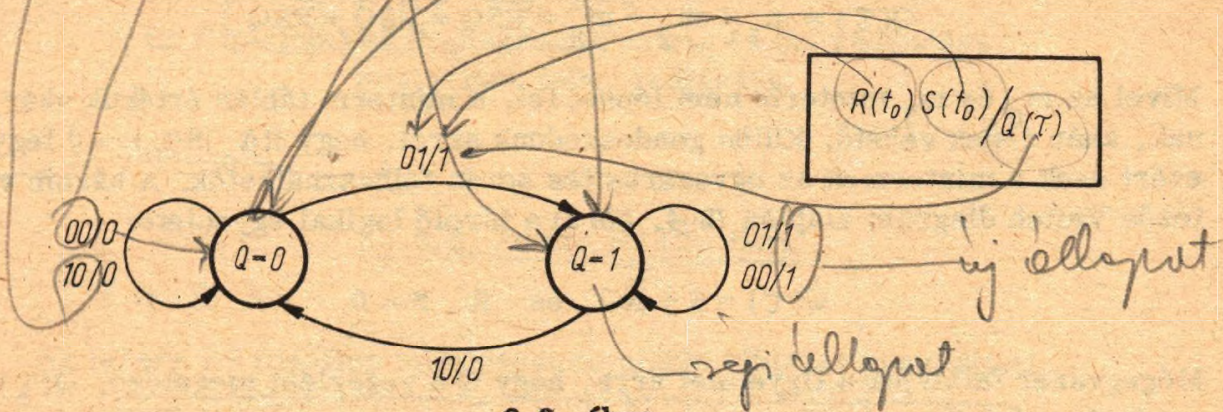
1. táblázat

$R=R(t_0)$	$S=S(t_0)$	$Q(\tau)=Q(t_0+\Delta t)$
0	0	$Q(t_0)$
0	1	1
1	0	0
1	1	Határozatlan

2. táblázat

index	$R=R(t_0)$	$S=S(t_0)$	$Q=Q(t_0)$	$Q(\tau)=Q(t_0+\Delta t)$	Flip-flop állapota
0	0	0	0	0	Változatlan
1	0	0	1	1	Változatlan
2	0	1	0	1	1-es állapotba billen
3	0	1	1	1	1-es állapotban marad
4	1	0	0	0	0-ás állapotban marad
5	1	0	1	0	0-ás állapotba billen
6	1	1	0	?	Határozatlan
7	1	1	1	?	Határozatlan

b) Állapotdiagram. A tároló két stabil állapottal rendelkezik, $Q=0$ ill. $Q=1$. A stabil állapotokat, valamint az átmenetet egyik állapotból a másikba grafikusán is ábrázolhatjuk (2-3. ábra). A stabil állapotokat körök, az átme-



2-3. ábra

RS flip-flop állapotdiagramja

netet körök közötti görbék ábrázolják, az átmenet irányának (nyíl) és feltételének (számok) feltüntetésével. A görbékre írt számok számlálója az $R(t_0)S(t_0)$ vezérlési feltételt, nevezője a vezérlés utáni $Q(\tau)$ állapotot reprezentálja. Az állapotdiagramon nem tüntettük fel az $RS = 11$ határozatlan állapotot eredm-

nyező vezérlést. Látható, hogy az állapotdiagram a vezérlési feltételekből közvetlenül felrajzolható. Az állapotdiagramot néha átmenetdiagramnak is nevezik.

c) Állapot (vezérlési) táblázat. Az állapot táblázatot általában az állapotdiagram alapján rajzoljuk fel. Az állapot táblázat felső vízszintes sora az RS lehetséges vezérlési kombinációkat, első oszlopa a $Q = Q(t_0)$ vezérlés előtti állapotot tartalmazza. Az állapot táblázatba a vezérlés utáni $Q(\tau)$ állapotot írjuk be:

Q	RS			
	00	01	10	11
0	Q	1	Q	?
1	Q	Q	0	?

Q \ RS	00	01	10	11
0	0	1	0	?
1	1	1	0	?

Az állapot táblázatot kétféleképpen is felrajoltuk. A változatlan állapotok, ill. a mindenkori állapot (0 vagy 1) feltüntetésével.

d) Tároló logikai egyenlete vagy karakterisztikus egyenlete. A tároló működését matematikai alakban pl. az igazságtáblázatból diszjunktív normál alakban a következőképpen írhatjuk fel:

$$Q(\tau) = m_1 + m_2 + m_3 = \bar{R}\bar{S}Q + \bar{R}S\bar{Q} + R\bar{S}Q$$

Mivel az m_6 és m_7 minterm nem léphet fel, a minterm táblán értékük akár 0-nak, akár 1-nek vehető. Külön gondoskodunk arról, hogy $R(t) \cdot S(t) = 0$ legyen, ezért ezek a mintermek az egyszerűsítés során felhasználhatók. A három változós Veitch diagram alapján (2-4. ábra) a tároló logikai egyenlete:

$$Q(\tau) = S + \bar{R}Q \quad \text{és} \quad R \cdot S = 0$$

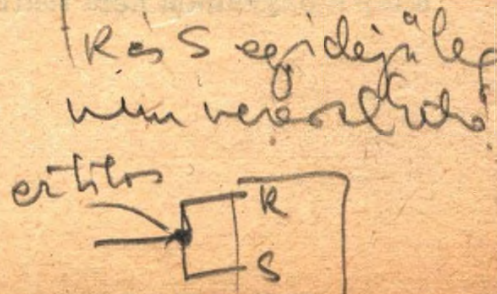
Mégegyszer felhívjuk a figyelmet arra, hogy Q a vezérlést megelőző, $Q(\tau)$ a vezérlés utáni állapot. Hasonlóan:

$$\bar{Q}(\tau) = R + \bar{S}\bar{Q} \quad \text{és} \quad R \cdot S = 0$$

e) Tároló logikai differencia egyenlete. Szinkron működésű szekvenciális hálózatokban a k -adik ütemben, a t_0 pillanatban a tároló állapotára ill. a vezérlőjelek értékére bevezethetjük a következő jelölést:

pl:

$$\begin{aligned} Q(\tau) &= \phi + \bar{\phi} \cdot \phi = \phi \\ Q(\tau) &= 1 + \bar{\phi} \cdot 1 = 1 \\ Q(\tau) &= \phi + 1 \cdot \phi = \phi \\ Q(\tau) &= \phi + \bar{1} \cdot \phi = \phi \end{aligned}$$



$$Q(t_0) = Q^k, \quad R(t_0) = R^k, \quad S(t_0) = S^k$$

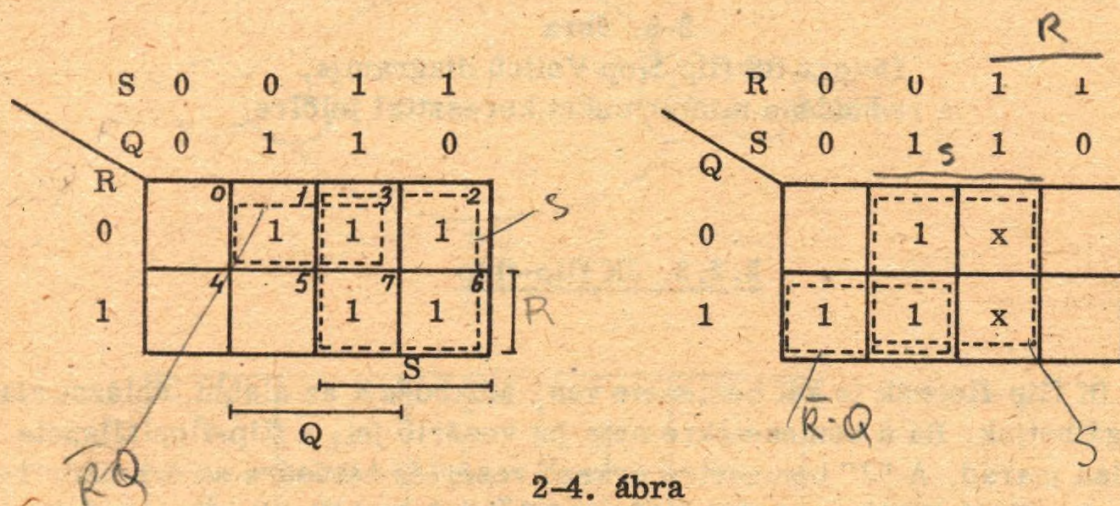
a vezérlés utáni állapotra pedig a:

$$Q(\tau) = Q(t_0 + \Delta t) = Q^{k+1}$$

jelölést. Ezekkel a jelölésekkel a tároló logikai egyenlete:

$$Q^{k+1} = S^k + \overline{R^k} Q^k \quad \text{és} \quad R^k \cdot S^k = 0$$

Mivel az egyenlet a k -adik és a $(k+1)$ -edik ütem között teremt kapcsolatot, a matematikai differencia egyenletek analógiájára logikai differencia egyenletnek nevezzük.



2-4. ábra
RS flip-flop Veitch diagramja az m_6 és m_7 redundáns mintermek felhasználásával

2.2.2. Inverz RS flip-flop

Az inverz RS flip-flop működését az alábbi táblázat szemlélteti:

$R=R(t_0)$	$S=S(t_0)$	$Q(\tau)=Q(t_0 + \Delta t)$
0	0	Határozatlan
0	1	1
1	0	0
1	1	$Q(t_0)$

Logikai egyenlete a 2-5. ábra Veitch (Karnaugh) diagramja alapján:

$$Q(\tau) = \bar{R} + SQ$$

	R	0	0	1	1
	S	0	1	1	0
Q					
0		x	1		
1		x	1	1	

2-5. ábra

Inverz RS flip-flop Veitch diagramja,
a redundáns mintermeket kereszttel jelölve

2.2.3. JK flip-flop

A JK flip-flopnak is két bemenete van. Működését az alábbi táblázat alapján vizsgálhatjuk. Ha a bemenetekre nem jut vezérlő jel, a flip-flop állapota változatlan marad. A "J" bemenetre érkező vezérlés hatására az áramkör 1-es állapotba kerül. A "K" bemenetre jutó vezérlőjel 0-ba billenti. Ha mindkét bemenetet egyidejűleg gerjesztjük, a flip-flop ellentétes állapotba billen.

index	$K=K(t_0)$	$J=J(t_0)$	$Q=Q(t_0)$	$Q(\tau)=Q(t+\Delta t)$	Flip-flop állapota
0	0	0	0	0	Változatlan
1	0	0	1	1	Változatlan
2	0	1	0	1	1-es állapotba billen
3	0	1	1	1	1-es állapotban marad
4	1	0	0	0	0-ás állapotban marad
5	1	0	1	0	0-ás állapotba billen
6	1	1	0	1	Ellentétes állapotba billen
7	1	1	1	0	Ellentétes állapotba billen

A JK flip-flop állapot diagramját és állapot táblázatát 2-6. ábra mutatja.
A tároló logikai egyenlete:

ill. $Q(\tau) = m_1 + m_2 + m_3 + m_6$

$$\bar{Q}(\tau) = m_0 + m_4 + m_5 + m_7$$

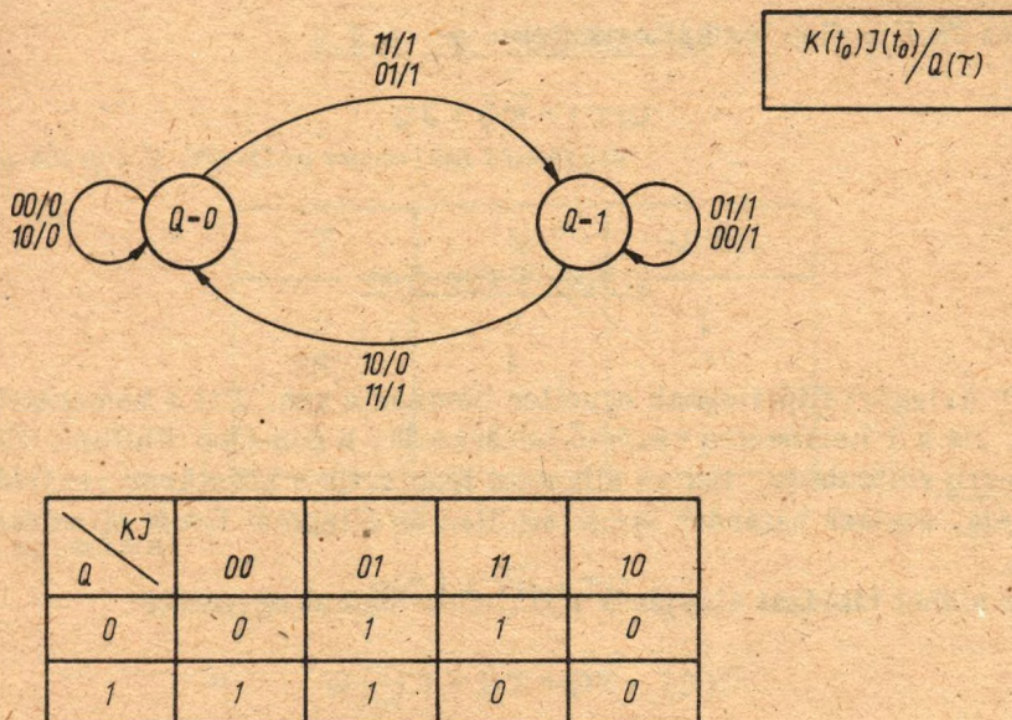
Egyszerűsítés után:

$$Q(\tau) = \bar{K}Q + J\bar{Q}$$

ill.

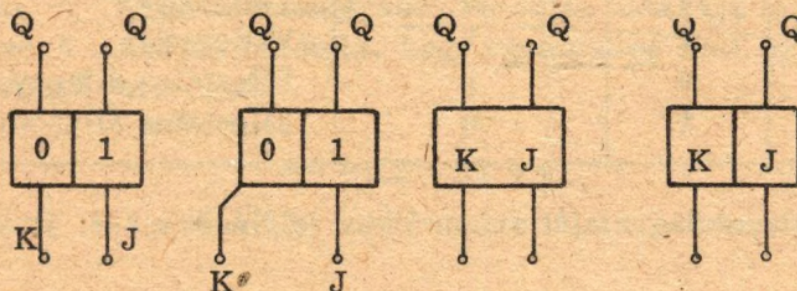
$$\bar{Q}(\tau) = KQ + \bar{J}\bar{Q}$$

JK flip-flop néhány szimbolikus ábrázolását 2-7. ábrán tüntettük fel.



2-6. ábra

JK flip-flop állapotdiagramja és állapottáblázata



2-7. ábra

JK flip-flop néhány szimbolikus ábrázolása

2.2.4. Inverz JK flip-flop

Az inverz JK flip-flop működését az alábbi táblázat szemlélteti

$K=K(t_0)$	$J=J(t_0)$	$Q(\tau)=Q(t_0 + \Delta t)$
0	0	$\bar{Q}$
0	1	1
1	0	0
1	1	Q

Az inverz JK flip-flop logikai egyenlete:

$$Q(\tau) = \bar{K}\bar{Q} + JQ$$

2.2.5. T flip-flop

A "T" (trigger) flip-flopnak egyetlen bemenete van. Ezt a bemenetet T-vel jelöljük. Ha a T bemenetre vezérlő jel érkezik, a flip-flop átbillen, állapotát ellentétesre változtatja. Ezt az állapotot fenntartja a következő vezérlőjel megérkezéséig, aminek hatására ismét átbillen és állapotát ismét ellentétesre változtatja.

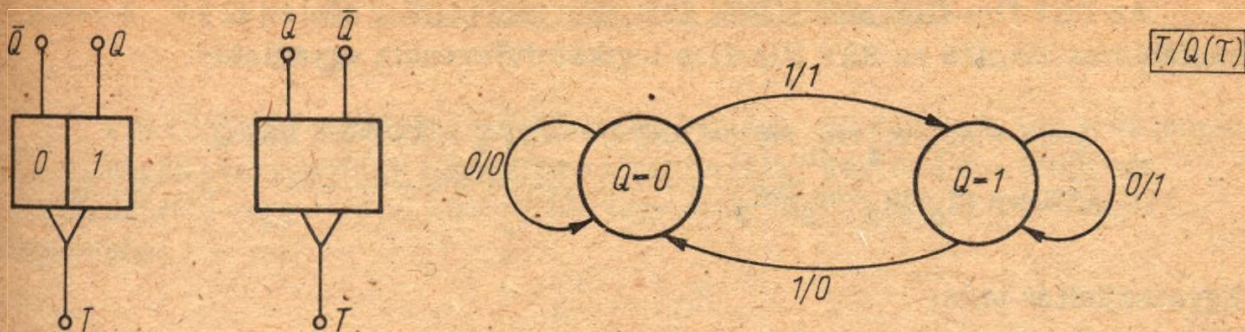
Az alábbi táblázat alapján a T flip-flop logikai egyenlete:

$$Q(\tau) = \bar{T}Q + T\bar{Q} = T \oplus Q$$

$$\bar{Q}(\tau) = TQ + \bar{T}\bar{Q} = T \odot Q$$

index	$T=T(t_0)$	$Q=Q(t_0)$	$Q(\tau)=Q(t_0 + \Delta t)$	Flip-flop állapota
1	0	0	0	Változatlan
2	0	1	1	Változatlan
3	1	0	1	Ellentétes állapotba billen
4	1	1	0	Ellentétes állapotba billen

A T flip-flop állapotdiagramját szimbolikus jelölését a 2-8. ábra illusztrálja.



2-8. ábra

T flip-flop szimbolikus ábrázolása és állapot diagramja

2.2.6. Inverz T flip-flop

Az inverz T flip-flop vezérlési táblázata:

T	Q	Q(τ)
0	0	1
0	1	0
1	0	0
1	1	1

Logikai egyenlete:

$$Q(\tau) = \overline{T}\overline{Q} + TQ$$

2.2.7. RST flip-flop

Az RST flip-flopnek három bemenete van (R,S,T). Az RS flip-flopnek megfelelően S bemeneten vezérelve 1-es állapotba, R bemeneten vezérelve 0 állapotba billen. A "T" bemeneten érkező vezérlés hatására mindig ellentétes állapotba billen. A működés feltétele, hogy egyidejűleg csak egyetlen bemenet-re érkezhets vezérlő jel. Tehát:

$$RS = RT = ST = 0$$

Az RST flip-flop működését az alábbi táblázatban tüntettük fel. Az igazságtáblázat alapján az RST flip-flop logikai differencia egyenlete:

$$Q(\tau) = m_1 + m_4 + m_5 + m_2 = \bar{R}\bar{S}TQ + \bar{R}S\bar{T}\bar{Q} + \bar{R}S\bar{T}Q + \bar{R}S\bar{T}\bar{Q}$$

$$\bar{Q}(\tau) = m_0 + m_8 + m_9 + m_3$$

Egyszerűsítés után:

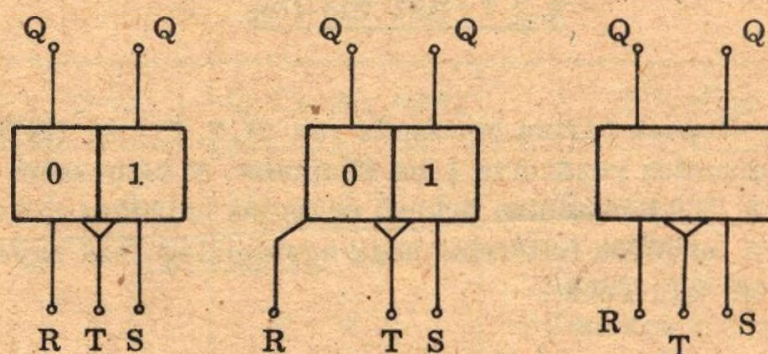
$$Q(\tau) = S + T\bar{Q} + \bar{R}TQ$$

ill.

$$\bar{Q}(\tau) = R + TQ + \bar{S}\bar{T}\bar{Q}$$

in-dex	R=R(t _o)	S=S(t _o)	T=T(t _o)	Q=Q(t _o)	Q(τ)=Q(t _o +Δt)	Flip-flop állapota
0	0	0	0	0	0	Változatlan
1	0	0	0	1	1	Változatlan
4	0	1	0	0	1	1-es állapotba billen
5	0	1	0	1	1	1-es állapotban marad
8	1	0	0	0	0	0-ás állapotban marad
9	1	0	0	1	0	0-ás állapotba billen
2	0	0	1	0	1	Ellentétes állapotba billen
3	0	0	1	1	0	Ellentétes állapotba billen

A 6, 7, 10, 11, 12, 13, 14, 15-ös indexű mintermekhez tartozó állapotok határozatlanok, az igazságtáblázatban ezeket nem tüntettük fel. Az RST flip-flop szimbolikus jelölését a 2-9. ábrán láthatjuk.



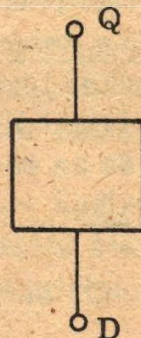
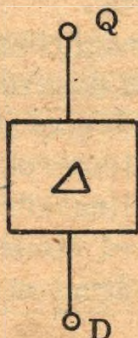
2-9. ábra
RST flip-flop szimbolikus jelölése

2.2.8. D flip-flop vagy késleltető elem

A D flip-flop vezérlési táblázatát és jelképi jelölését a 2-10. ábra mutatja. Szinkron szekvenciális hálózatban alkalmazzák, késleltetési ideje két szinkronizáló vagy órajel közötti idő (egy ütem idő). Logikai differencia egyenlete:

$$Q^{k+1} = D^k$$

D^k	Q^{k+1}
0	0
1	1



2-10. ábra

D flip-flop vezérlési táblázata és szimbolikus jelölése

2.2.9. A tárolók logikai egyenleteinek összefoglalása

A tárolók logikai egyenleteit összefoglalóan az alábbi táblázat tartalmazza:

Tároló típusa	Logikai egyenlete	Megjegyzés
RS	$Q(\tau) = S + \bar{R}Q$ és $RS = 0$	
JK	$Q(\tau) = \bar{K}Q + J\bar{Q}$	J = K = T esetén T flip-flop
T	$Q(\tau) = \bar{T}Q + T\bar{Q}$	
RST	$Q(\tau) = S + T\bar{Q} + \bar{R}TQ$	R=S=0... esetén T flip-flop T=0... esetén RS flip-flop
D	$Q^{k+1} = D^k$	

2.3 Tároló áramkörök realizálása

2.3.1. Bevezetés

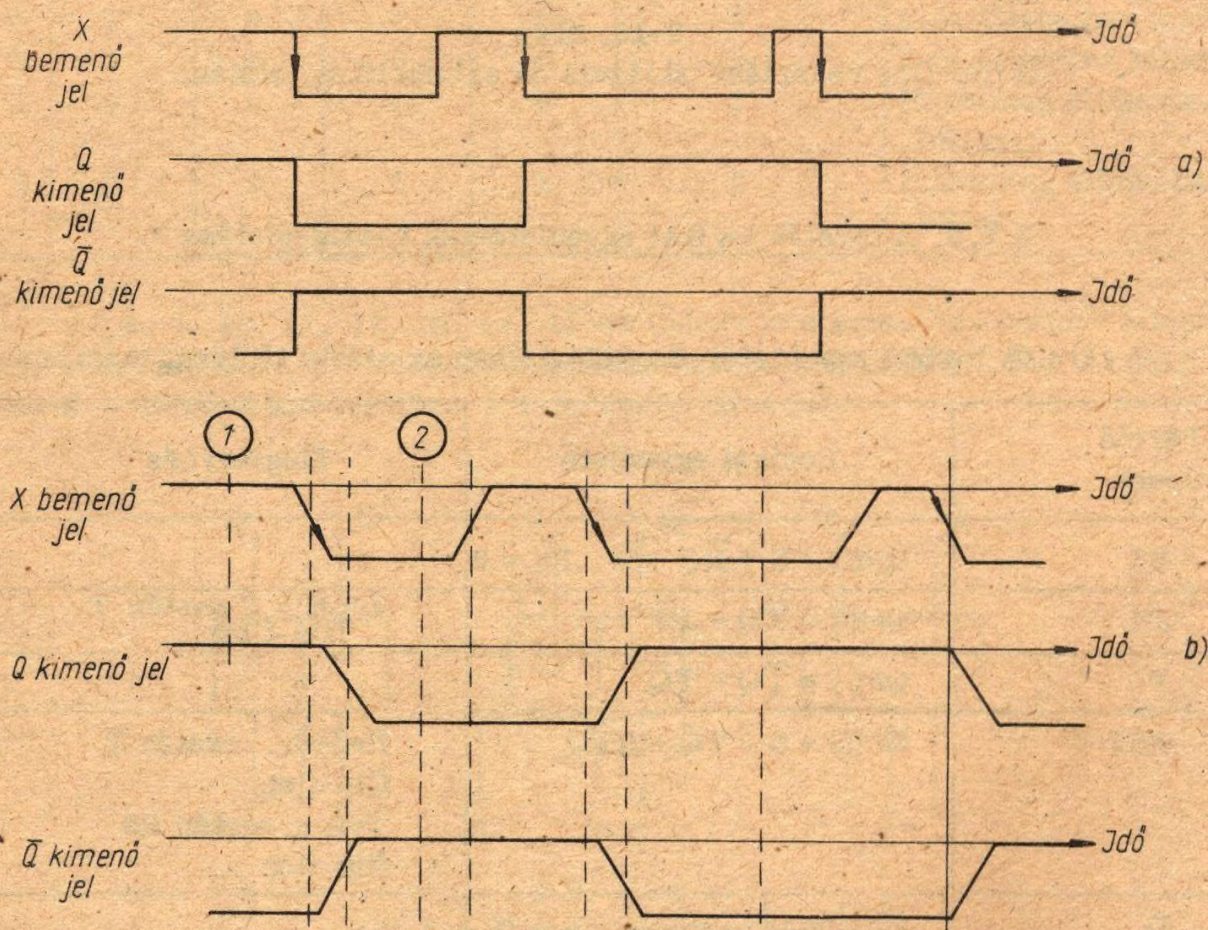
A tároló áramkörök vezérlése történhet

- a) statikus szintekkel.
- b) vezérlőjel $0 \rightarrow 1$ ill. $1 \rightarrow 0$ átmeneteivel.

Ennek megfelelően a tároló áramkörök vezérlésük alapján lehetnek statikus és dinamikus vezérlésű áramkörök.

Statikus vezérlésű billenőköröket statikus logikai szintekkel vezéreljük. A vezérlőjelnek legalább olyan hosszú ideig fenn kell maradnia, amíg a billentés be nem fejeződik. A billentéssel járó tranziens idő egy felfutási, egy lefutási időből és az esetleges töltéstárolás idejéből tevődik össze.

Dinamikus vezérlésű billenőkörök átbillenését a vezérlőjel $0 \rightarrow 1$ ill. $1 \rightarrow 0$ átmenete idézi elő. A $0 \rightarrow 1$ átmenet negatív logikai szint esetén negatív impulzus megjelenését, $1 \rightarrow 0$ átmenet pedig pozitív impulzus megjelenését jelel.



2-11. ábra

Aszinkron bináris osztó jelalakjai
a) ideális esetben, b) valóságos esetben

A vezérlőjelek változása véges idő alatt megy végbe tisztán sztatikus vezérlésű aszinkron szekvenciális hálózatokban is. A viszonyok tanulmányozására ábrázoljuk egy statikus, logikai szintekkel működő aszinkron bináris osztó kimenő és bemenő jelalakjait (2-11. ábra) ideális és valóságos esetben. Mivel a kimenőjel időkésséssel követi a bemenőjel változásait, az 1 és 2-vel jelölt stabil állapotok között instabil állapotok jönnek létre.

Attól függően, hogy szinkronizáló jelekkel, órajelekkel szinkronizáljuk-e a tároló működését, megkülönböztetünk

- a) órajellel szinkronizált vagy röviden szinkronizált
- b) szinkronizáló impulzus nélküli, vagy röviden aszinkron

tárolókat. Szinkronizált működésű tárolók állapotváltozása a szinkronizáló impulzus beérkezésének pillanatában következhet csak be. Aszinkron rendszerekben az állapotváltozások tetszőleges időpontokban jöhetnek létre.

2.3.2. Sztatikus vezérlésű aszinkron RS tároló

Ismeretes az univerzális logikai műveleteknek az a tulajdonsága, hogy segítségükkel minden logikai függvény megvalósítható. Valósítsuk meg a Peirce-féle (NVAGY) művelettel az RS tárolót. A realizálás a tároló elektromos kapcsolási rajzának megtervezéséhez vezet.

A realizálandó RS flip-flop logikai egyenlete az igazságtáblázat alapján:

$$Q(\tau) = m_1 + m_2 + m_3 = \bar{R}\bar{S}Q + \bar{R}S\bar{Q} + \bar{R}SQ \quad \text{lásd 11. old.}$$

Az m_6 és m_7 -es mintermek nem léphetnek fel, mivel R és S egyidejűleg nem vezérelhető. Ezért m_6 és m_7 együtthatója akár 0-nak, akár 1-nek vehető. Értéküket úgy választjuk meg, hogy a realizálandó logikai hálózat a legegyszerűbb legyen.

Mivel a $Q(\tau)$ egyenletet NVAGY kapuk segítségével kívánjuk realizálni, a szokásos jelöléssel ez azt jelenti, hogy NV [NV (1_M)] alakban kell $Q(\tau)$ -t előállítani. Tehát realizáláshoz a maxterm tábla 1-es termjeit kell kiolvasnunk. Ábrázoljuk $Q(\tau)$ -t maxterm táblán, figyelembe véve, hogy a maxterm táblán ott van 1, ahol a minterm táblán zérus (2-12. ábra).

A maxterm táblán a két szabadon választható term együtthatóját 1-nek választva, $Q(\tau)$ kétszintű realizálását a 2-13. ábrán láthatjuk. A bemeneten levő alsó NVAGY kapu elhagyható, mivel eleve az R vezérlőjel áll rendelkezésünkre. Az egyszerűsített logikai hatásvázlatot is a 2-13. ábrán fűntettük fel.

A tárolót úgy kell felépíteni, hogy a tárolt információ és annak invertált értéke is rendelkezésünkre álljon. Ezért ábrázoljuk $\bar{Q}(\tau)$ -t is maxterm táblán. A szabadon választható termeket itt is kérdőjellel jelöltük (2-14. ábra). A szabadon választható együtthatókat itt is 1-nek választva, $\bar{Q}(\tau)$ kétszintű realizá-

avagy NOR-hoz

QR

S	0	0	1	1
Q	0	1	1	0
R	0	1	1	0
0		1	1	1
1			?	?

MINTERM TÁBLA

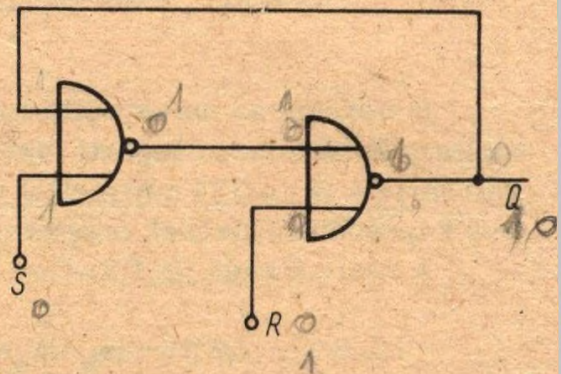
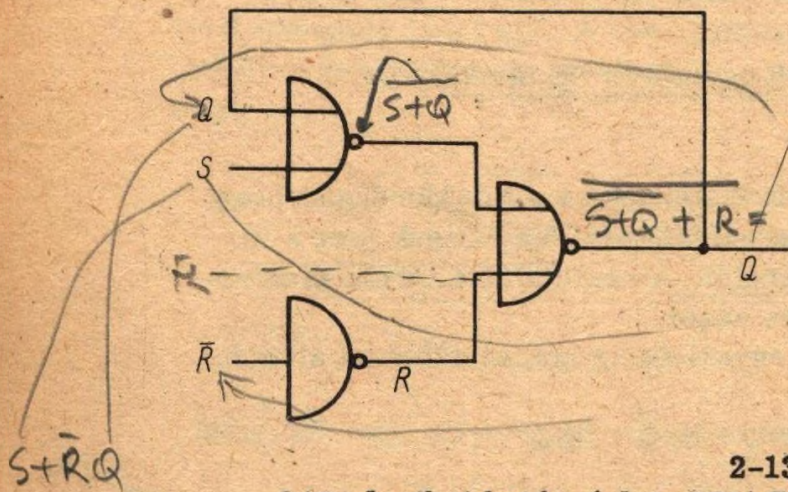
S	1	1	0	0
Q	1	0	0	1
R	1	0	0	1
1	1			
0	1	1	?	?

MAXTERM TÁBLA

2-12. ábra

$Q(\tau) = m_1 + m_2 + m_3$ ábrázolása minterm és maxterm táblán

$$\bar{R} + (S + Q)$$



2-13. ábra

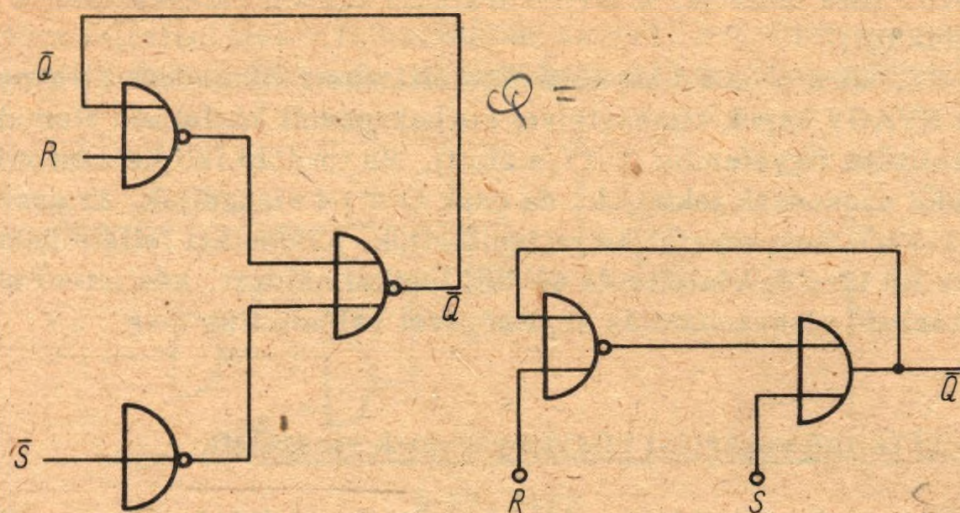
$Q(\tau)$ logikai hatásvázlata NVAGY kapukkal történő realizálás esetén

S	1	1	0	0
Q	1	0	0	1
R				
1		1	1	1
0			?	?

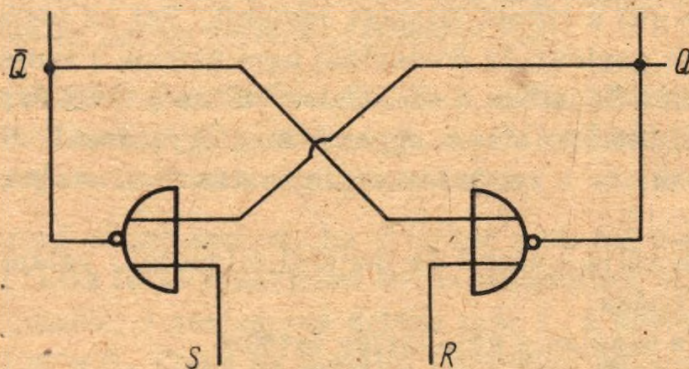
MAXTERM TÁBLA

2-14. ábra

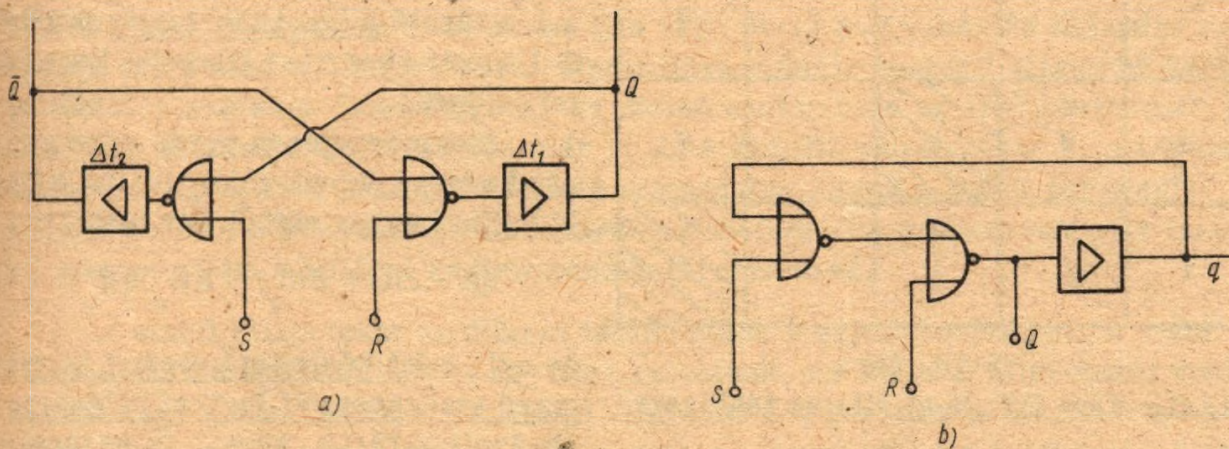
$Q(\tau)$ ábrázolása maxterm táblán



2-15. ábra
 $Q(\tau)$ realizálása NVAGY kapuk segítségével



2-16. ábra
 Aszinkron RS flip-flop realizálása Peirce féle univerzális logikai műveletekkel



2-17. ábra
 RS flip-flop logikai hatásvázlata az időkéésések figyelembevételével

lását a 2-15. ábra mutatja. A tároló áramkör logikai hatásvázlatát a 2-16. ábrán tüntettük fel.

A 2-16. ábra NVAGY kapuinak késleltetéseit állandónak és koncentrálnak tekintve a NVAGY kapuk kimeneteivel sorbakapcsolt késleltető elemekkel vehetjük közelítően figyelembe (2-17/a ábra). Ha további feltételként mindkét késleltetési időt azonosnak tekintjük, és csak $Q(\tau)$ -t vizsgáljuk, az aszinkron RS tárolót a 2-17/b ábra szerint helyettesíthetjük. Kiindulási feltételünknek megfelelően az RS tárolót késleltetés nélküli logikai hálózat, késleltető elem és az ezekre kiterjedő visszacsatolás segítségével valósítottuk meg.

2.3.2.1. RS tároló vezérlési tulajdonságainak vizsgálata

Mielőtt az RS tároló kapcsolási rajzát felrajzolnánk, elemezzük a tároló működését. A tároló stabil állapotban van, ha:

$$Q(t) = q(t)$$

Amennyiben $Q(t) \neq q(t)$ a tároló állapota instabil. Ezt az instabil állapotot tehát az jellemzi, hogy a gerjesztés (vezérlés) nem azonos a kimeneti változó (általános szekvenciális hálózatban a szekunder változó) értékével, vagyis a késleltető elem be- és kimenő jele nem egyezik meg egymással. Ez az állapot nem marad fenn tartósan, és a rendszer a gerjesztés által meghatározott állapotba jut.

Vizsgálatunk célja a stabil és instabil állapotok, valamint az ezek közötti átmenet meghatározása.

A 2-17. ábra alapján a gerjesztés értéke

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

q \ R \ S	0	0	1	1
	0	1	1	0
0	0	1	x	0
1	1	1	x	0

2-18. ábra

RS flip-flop $Q(R, S, q)$ állapot táblázata

Veitch diagramon ábrázolva (2-18. ábra) a $Q(R, S, q)$ függvényt meghatározzuk a tároló állapot táblázatát (az $R=1$, $S=1$ tiltott vezérlési állapotot keresztrel jelöltük).

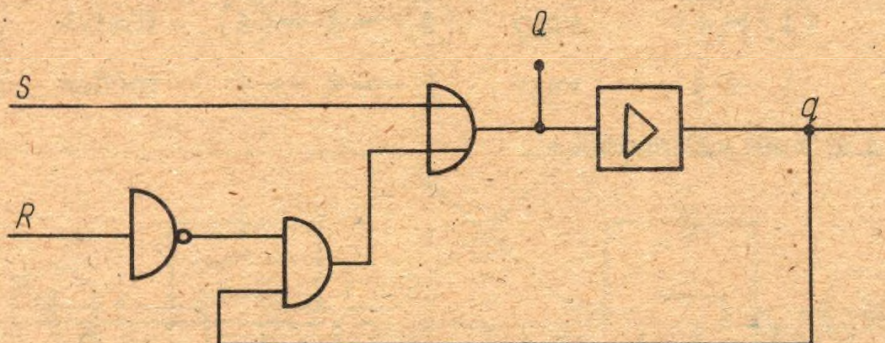
Természetesen ugyan erre az eredményre jutunk akkor is, ha a tároló logikai egyenletéből indulunk ki:

$$q(t + \Delta t) = S(t_0) + \overline{R}(t_0)q(t_0)$$

és ezt, mint késleltető elemet tartalmazó visszacsatolt logikai hálózatot ábrázoljuk (2-19. ábra). (Visszacsatolás során a függő változót a kombinációs áramkör független (bemenő) változójává tesszük.) Mivel a tároló vezérlése előtt stabil egyensúlyi állapotban van, ezért: $q(t_0) = Q(t_0)$; a vezérlés hatására a gerjesztés:

$$Q = S + \bar{R}q$$

A 2-19. ábra felrajzolásakor felhasználtuk az $R=1$, $S=1$ tiltott kombinációt, ezért szerepel a kapott kifejezés bal oldalának első tagjában $S, \bar{R}S$ helyett.



2-19. ábra.
RS tároló logikai hatásvázlata

Az állapot táblázatból látható, hogy amikor a tároló 0-ból 1-es illetve 1-ből 0-ás állapotba billen a gerjesztés és a kimenő változó nem egyezik meg egymással (vonalkázott termék) (l. a tároló igazság táblázatát is). A tároló működtetésében három esetet különböztethetünk meg:

a) Egyik bemenetet sem vezéreljük, $R=0$, $S=0$. Láthatóan a tároló előző állapotától függetlenül stabil állapotban van, a táblázat bal oldalán levő kimenő változó mindkét esetben azonos a Q gerjesztéssel.

b) SET bemenet vezérlésekor, ha előzőleg a tároló 0-ás állapotban volt, először az állapot tábla felső vízszintes során mozdulunk el. $R=0$, $S=1$ esetén instabil állapotba jutunk, a gerjesztés értéke már 1 a késleltetés miatt a kimeneti változó (q) még 0. Ebből az instabil állapotból ugyanazon oszlopban lefelé elmozdulva a $q=1$ -nek megfelelő stabil egyensúlyi állapotba jutunk.

Ha a tároló eredetileg 1-es állapotban volt, a vezérlés hatására (vízszintes irányban elmozdulva) 1-es állapotban marad.

c) RESET bemenet vezérlésekor hasonlóan analizálható a hálózat. Ha pl. a tároló $R=0$, $S=0$ esetén 1-es állapotban volt, $R=1$, $S=0$ vezérlés hatására az állapot tábla alsó vízszintes során elmozdulva először instabil állapotba kerül. (A táblák a széleken összefüggnek, az utolsó oszlop után az első következik.) Ebben az esetben a Q gerjesztés már 0, de a kimenő változó még 1. Az instabil állapotból ugyanazon oszlopban felfelé elmozdulva jutunk a stabil egyensúlyi állapotba, ahol a gerjesztés és a kimenő változó értéke megegyezik. Az instabil állapot idejét a hálózat késleltetése határozza meg.

A stabil állapotokat az állapot táblán külön megjelölhetjük, pl. a stabil állapotoknak megfelelő helyre egy kört rajzolunk, az instabil állapotokat vonalkázzuk vagy ponttal jelöljük. A stabil állapotba a Q gerjesztés beírását el is hagyhatjuk, mivel annak értéke megegyezik a kimenő változó értékével. A könnyebb tárgyalás érdekében a különböző állapotokat számozással vagy betűk segítségével különböztetjük meg egymástól. Az azonos oszlopban végbemenő

instabil $\rightarrow$ stabil átmeneteket az átmenet irányát is feltüntető egyenesekkel je-
lölhetjük (2-20. és 2-21. ábra). Ily módon pl. $R=1, S=0$ vezérlés esetén

pl. $(A) \rightarrow (D)$ vagy $(B) \rightarrow D \rightarrow (D)$ illetve
 $(1) \rightarrow (4)$ vagy $(2) \rightarrow 4 \rightarrow (4)$ illetve
 $\underline{1} \rightarrow \underline{4}$ vagy $\underline{2} \rightarrow 4 \rightarrow \underline{4}$ módon

jelölhetjük az állapot változásokat.

q \ R S	0 0	0 1	1 1	1 0
	0	1	0	1
0	○		x	○
1	○	○	x	

q \ R S	0 0	0 1	1 1	1 0
	0	1	0	1
0	○	▨	x	
1			x	▨

q \ R S	0 0	0 1	1 1	1 0
	0	1	0	1
0	○	○ ↓	x	○ ↑
1	○	○ ↑	x	○ ↓

2-20. ábra

RS tároló állapot táblázatának egyszerűsített jelölése

q \ R S	0 0	0 1	1 1	1 0
	0	1	0	1
0	○ 1	○ 3 ↓	x	○ 4 ↑
1	○ 2	○ 3	x	○ 4 ↑

q \ R S	0 0	0 1	1 1	1 0
	0	1	0	1
0	○ A	○ C ↓	x	○ D ↑
1	○ B	○ C ↑	x	○ D ↓

2-21. ábra

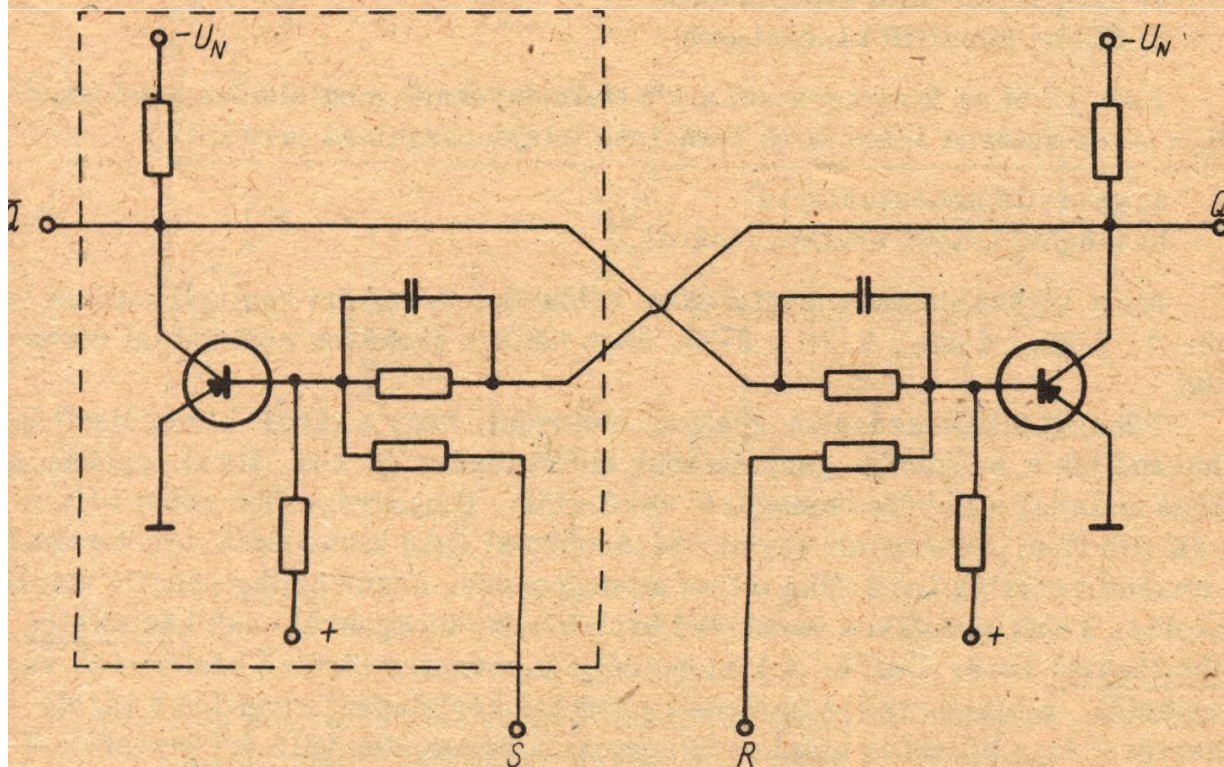
RS tároló állapotainak szimbolikus jelölése

Már most megjegyezzük, hogy a bemenő változók megváltozásának hatá-
sára az állapot változás mindig azonos sorban, instabil $\rightarrow$ stabil átmenetnél
pedig a változás valamely oszlopban történik.

2.3.2.2. Aszinkron működésű RS tárolók áramköri megvalósítása

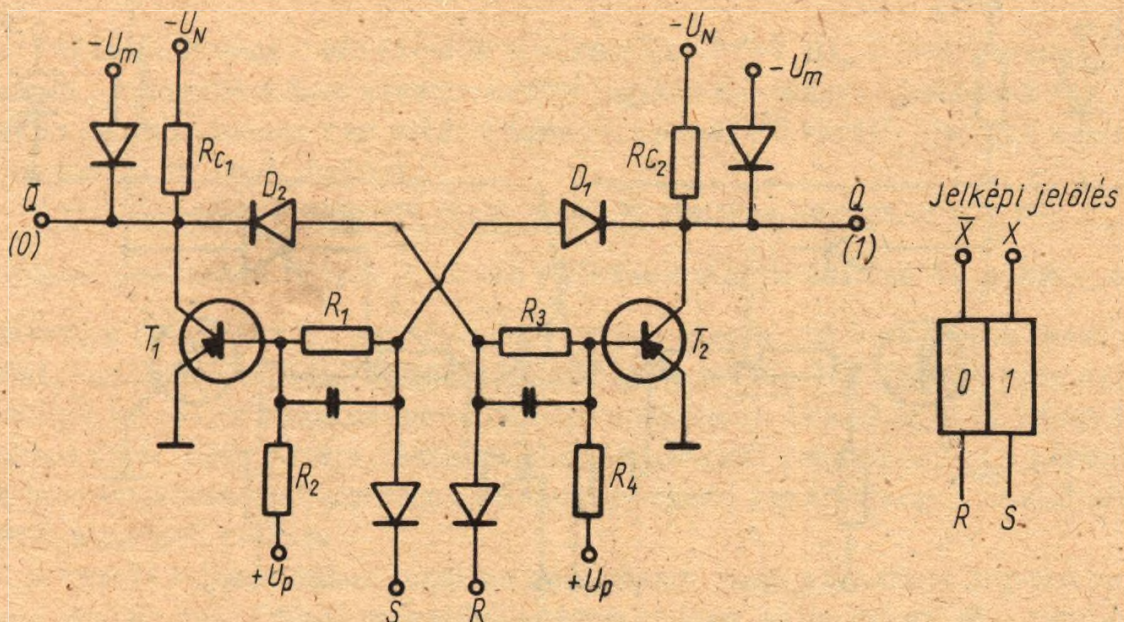
A logikai tervezés után felrajzolhatjuk az RS flip-flop kapcsolási rajzát.
 • A NVAGY műveletet megvalósíthatjuk pl.

- Ellenállás-inverteres logikai áramkörrel
- Dióda-inverteres logikai áramkörrel.



2-22. ábra

RS flip-flop megvalósítása szabad jelszintű, gyorsító kondenzátort alkalmazó, ellenállás-inverteres kapcsolással



2-23. ábra

Statikus vezérlésű tároló RS tároló megvalósítása kötött jelszintű, gyorsító kondenzátort alkalmazó dióda-inverteres kapcsolással

Az egyes inverterek lehetnek

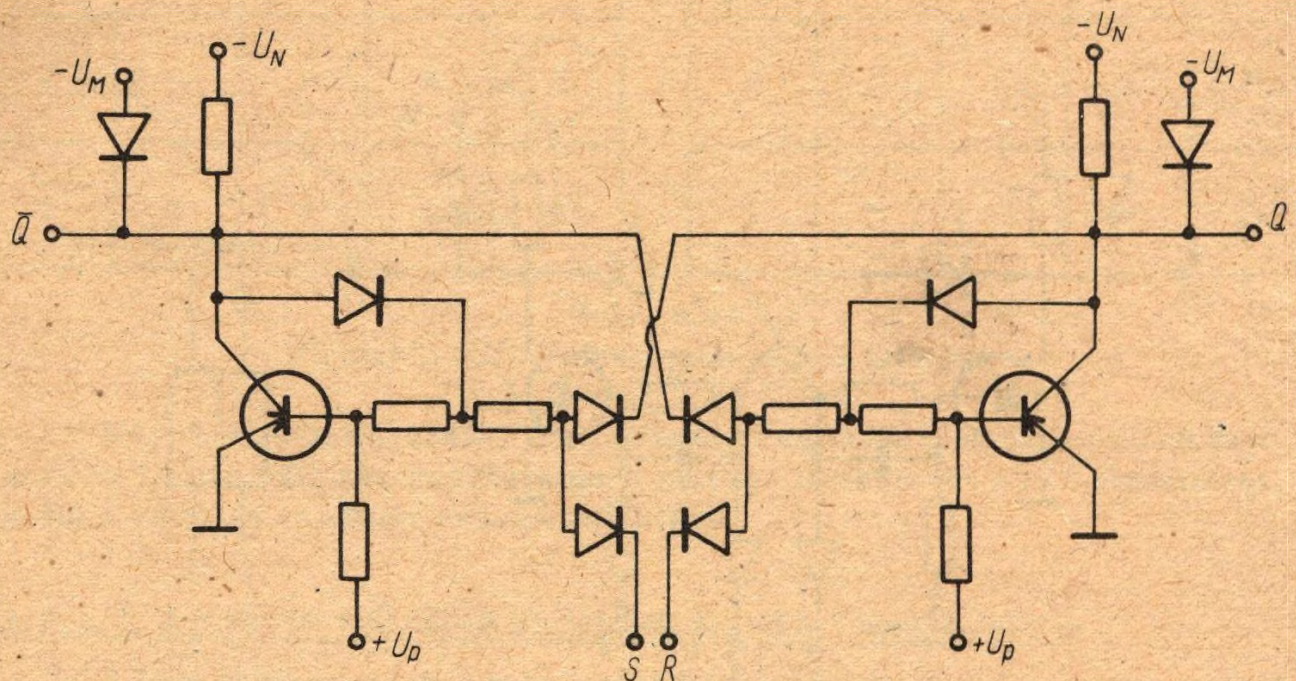
- a) Szabad jelszintű kapcsolások
- b) Kötött jelszintű kapcsolások

Ezen kívül az invertereknél a követelményeknek megfelelően gondoskodhatunk a tranziszoros áramkörök tranziens tulajdonságainak javításáról

- a) gyorsító kondenzátorral
- b) megfogó dióda alkalmazásával.

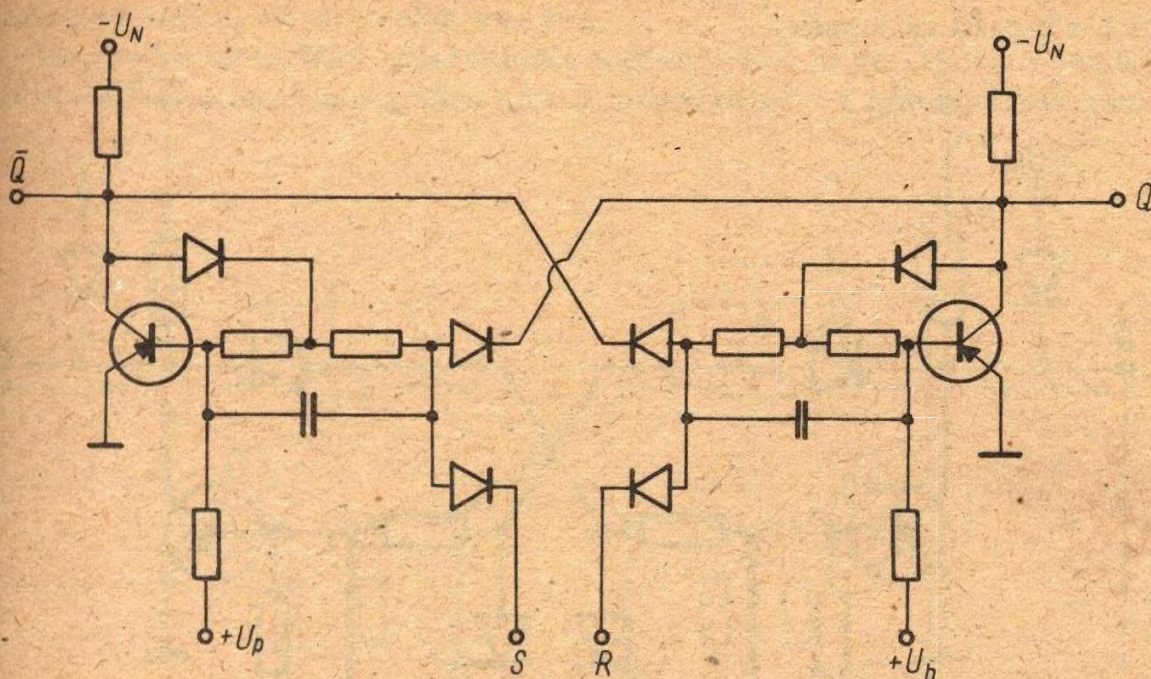
Igy a kívánalmaknak megfelelően különböző áramköri realizálásokhoz jutunk. A 2-22., 2-23., 2-24., 2-25. ábra néhány szokásos realizálási módot mutat.

Tárolók alkalmazásakor gyakran előfordul, hogy a tároló kimenőjelét is felhasználjuk a szekunder gerjesztések előállítására (l. 1-4. ábra), szokásos szóhasználat a tárolót önmagáról vezéreljük. (Legegyszerűbb példa erre a T, JK flip flop.) A tárgyalt aszinkron tárolóknál $\bar{Q}(Q)$ kimenetről S(R) bemenetre történő vezérlés logikailag teljes ellentmondás, áramkörileg negatív visszacsatolás. Tehát a statikus vezérlésű áramkörök önmagukról csak késleltetés beiktatásával vezérelhetők. A kimenetről a bemenetre történő késleltetett vezérelhetők. A kimenetről a bemenetre történő késleltetett vezérlés a tároló billentését és a billentés utáni új állapotnak a bemeneten való érvényesülését időben különválasztja. Ily módon lehetőség van logikai visszacsatolás alkalmazására. A tárolók önmagukról történő vezérlése külön késleltető elem beiktatása nélkül is megvalósítható dinamikus vezérlésű tárolók alkalmazásával.



2-24. ábra

RS flip-flop megvalósítása kötött jelszintű, megfogó diódás dióda-inverteres kapcsolással



2-25. ábra

RS flip-flop megvalósítása szabad jelszintű, megfogó diódás és gyorsító kondenzátort alkalmazó dióda-inverteres kapcsolással

2.3.3. Dinamikus vezérlésű tárolók

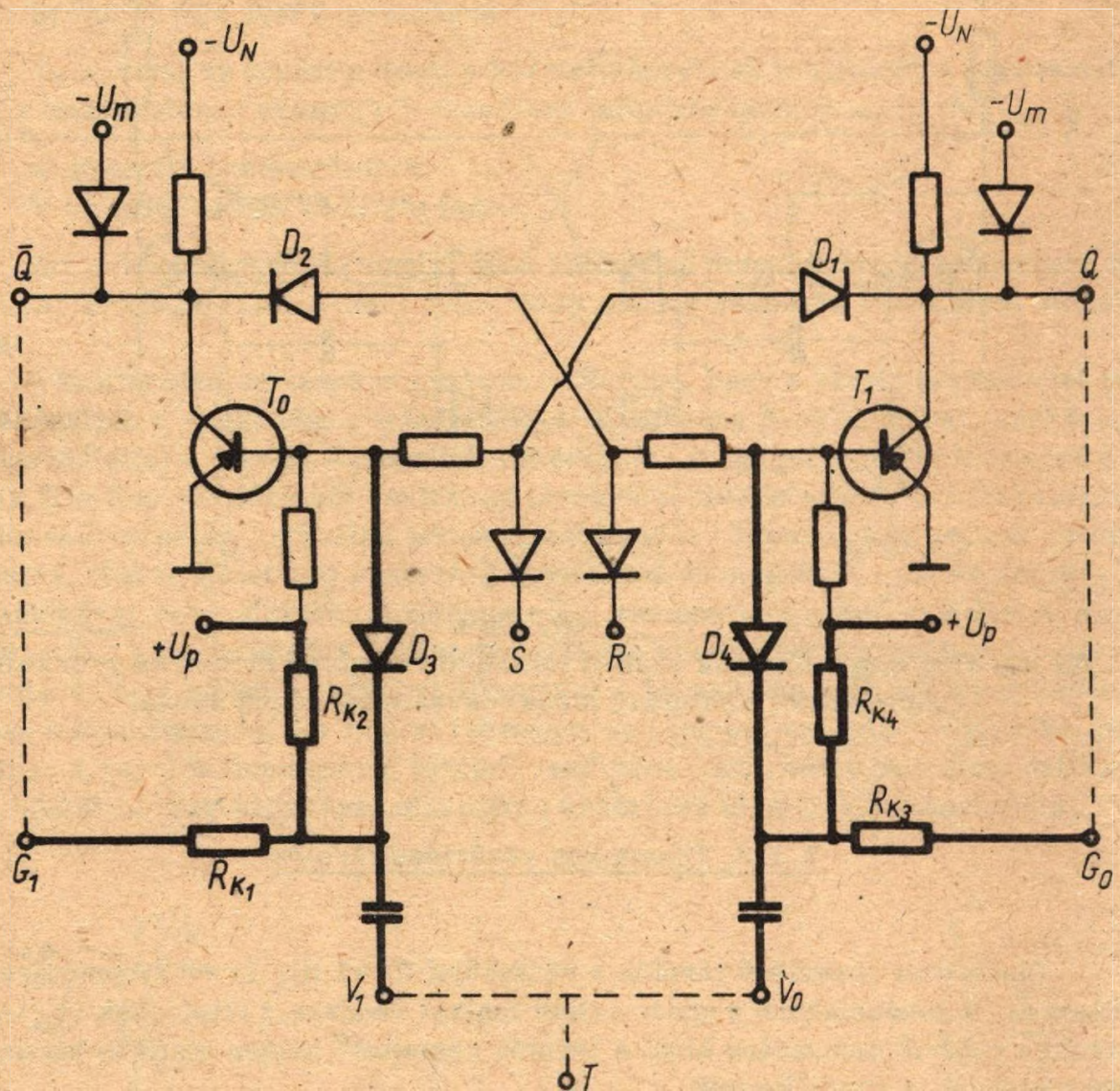
Dinamikus vezérlésű tárolók a vezérlőjel $0 \rightarrow 1$ ill. $1 \rightarrow 0$ átmenetére billennek. A továbbiakban negatív szintű logikai rendszert tételezünk fel. A dinamikus vezérlő bemeneten kívül a tárolók késleltető hatású statikus kapuzó bemenetekkel is rendelkeznek.

$0 \rightarrow 1$ átmenetre (negatív élre) működő statikus kapuzású (G_0 ill. G_1) és dinamikus vezérlésű (V_0 , V_1) tároló szokásos megvalósítása és jelképi jelölése látható a 2-26. ábrán. Az áramkört azon a dinamikus bemeneten lehet vezérelni, amelyhez tartozó statikus bemenőpontra előzőleg logikai Igen szintű vezérlőjel jutott. A statikus bemenőpontról nézve az áramkör R-C integráló elemként viselkedik, ezért késleltető tulajdonsággal rendelkezik. A kapuzó statikus bemenőjelnek a dinamikus bemenetre érkező vezérlőjelet a késleltetés idejével meg kell előznie.

A tároló billentéséhez tehát az szükséges, hogy a statikus kapuzó bemenetre a késleltetési idővel megelőzően logikai Igen szint érkezzék, a dinamikus bemeneten pedig a jel 0-ról 1-re változzék.

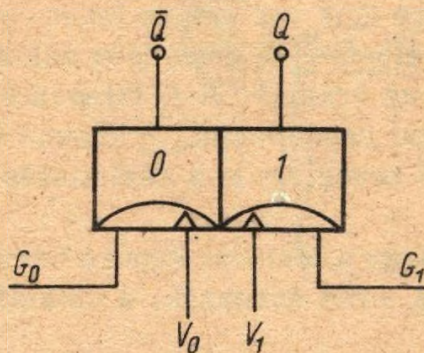
$1 \rightarrow 0$ átmenetre (pozitív élre) működő statikus kapuzású és dinamikus vezérlésű tároló szokásos áramköri megoldását mutatja a 2-27. ábra. A tároló billentéséhez az szükséges, hogy a statikus kapuzó bemenetre a késleltetési

idővel megelőzően logikai Nem szint érkezzék, a dinamikus bemeneten pedig a jel 1-ről 0-ra változzék.



2-26/a ábra

Dinamikus vezérlésű bistabil multivibrátor



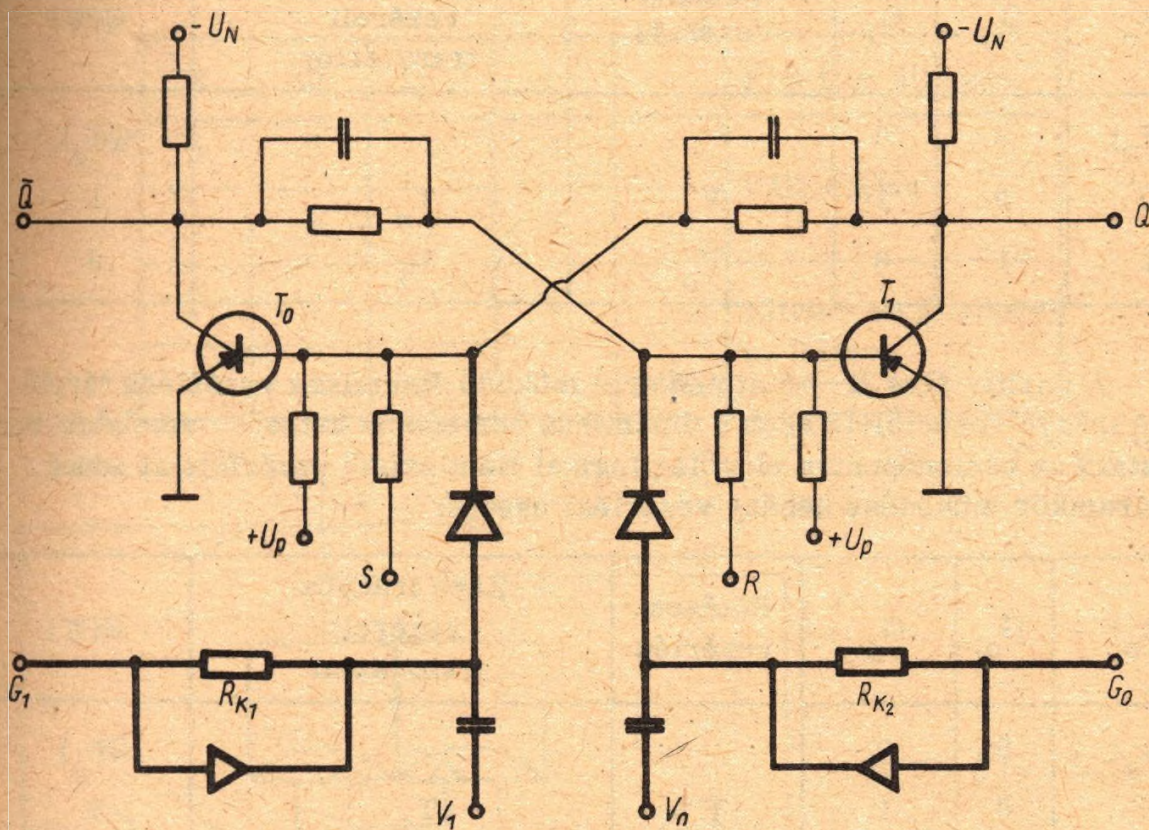
2-26/b ábra

Dinamikus vezérlésű tároló
jelképi jelölése

A dinamikus vezérlésű tárolók kimenetéről tehát a tárolót önmagáról is vezérelhetjük, ha a visszacsatolt jelet a késleltett hatásu (integráló tulajdonságu) kapuzó bemenetekre vezetjük. Összetett logikai hálózatokban a tárolók kimenetei felhasználhatók bármely fokozat bemenetének vezérlésére, mivel a dinamikus bemenetre érkező jel (órajel) beérkezésekor a tárolók kimenő jele nem változik. Statikus vezérlésű tároló kimenő jele a vezérlés pillanatában gyakorlatilag már változik, ezért szükséges

a bemenetre történő visszacsatolásakor a tároló tranziens idejénél nagyobb késleltetésű késleltető elem alkalmazása.

Dinamikus vezérlésű, késleltetett kapuzású tároló visszacsatolása tehát külön késleltető elemek beiktatása nélkül megengedett. A kombinációs hálózat



2-27. ábra

Dinamikus vezérlésű tároló ellenállás-inverter kapcsolás felhasználásával

áramköri visszacsatolást nem hozhat létre, kizárólag logikai visszacsatolást, amit legegyszerűbben dinamikus vezérlésű tárolók alkalmazásával valósíthatunk meg.

2.3.4. Dinamikus vezérlésű tárolók alkalmazása

Dinamikus vezérlésű tárolók segítségével valósíthatjuk meg a visszacsatolást tartalmazó különböző típusú flip-flop áramköröket.

2.3.4.1. Dinamikus RS flip-flop

A negatív élre ($0 \rightarrow 1$ átmenetre) működő tároló nyitó irányu vezérlőjele azon a dinamikus bemeneten hatásos, amelyhez tartozó statikus bemenőpontra

előzőleg logikai Igen szintű vezérlőjelet adunk. Az áramkör működését az alábbi táblázatban foglalhatjuk össze:

$Q(t_0)$	G_0	G_1	Hatásos vezérlés	Nyitó irányban vezérelt tranzisztor	$Q(\tau)$
$Q(t_0)$	0	0	-	-	$Q(t_0)$
0	0	1	V_1	T_0	1
1	1	0	V_0	T_1	0

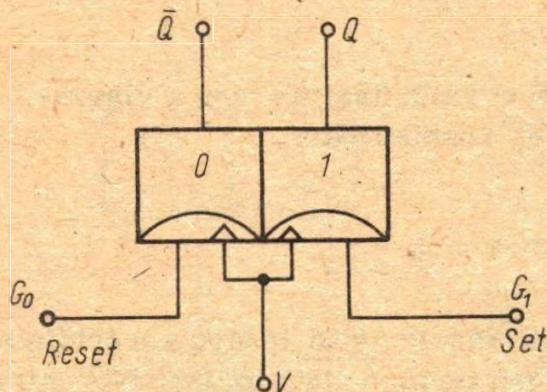
A pozitív élre ($1 \rightarrow 0$ átmenetre) működő dinamikus vezérlésű tároló záró irányu vezérlőjele azon a dinamikus bemeneten hatásos, amelyhez tartozó statikus bemenőpontra előzőleg logikai Nem szintű vezérlőjelet adunk. Az áramkör működése néhány vezérlési esetre:

$Q(t_0)$	G_0	G_1	Hatásos vezérlés	Záró irányba vezérelt tranzisztor	$Q(\tau)$
$Q(t_0)$	0	0	-	-	$Q(t_0)$
0	0	1	V_0	T_1	1
1	1	0	V_1	T_0	0

Láthatóan a dinamikus vezérlésű tároló RS flip-flop tulajdonságaival rendelkezik, de vezérléséhez először a kapuzó statikus bemeneteket, ezt követően az összeköthető dinamikus bemenetet kell vezérelni (2-28. ábra).

A dinamikus vezérlésű tároló logikai egyenlete:

$$Q(\tau) = G_1 + G_0 Q$$



G_0	G_1	$Q(\tau)$	Flip flop állapota
0	0	$Q(t_0)$	Változatlan
0	1	1	1-be billen
1	0	0	0-ba billen
1	1	?	Határozatlan

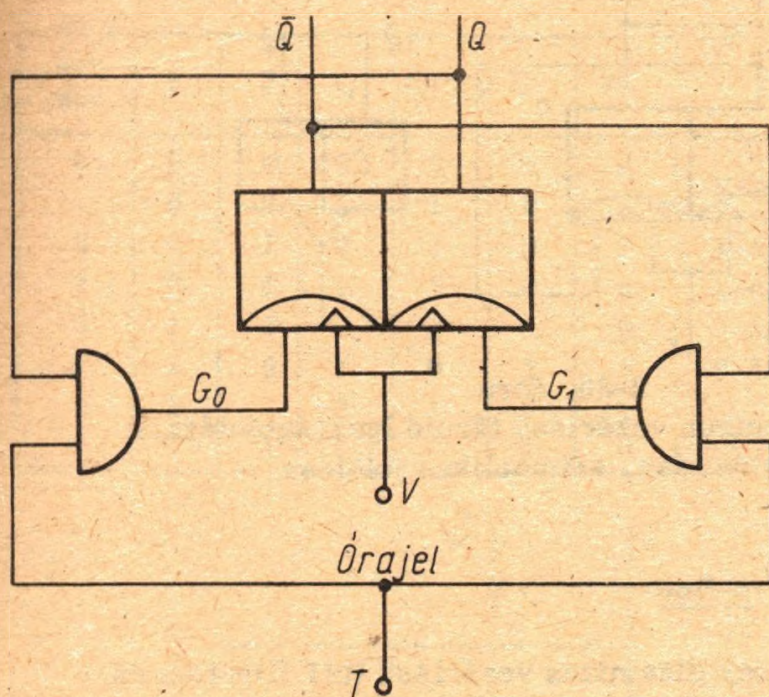
2-28. ábra

Statikus kapuzású, dinamikus vezérlésű tároló jelképi jelölése és igazságtáblázata

2.3.4.2. Dinamikus T flip-flop

T flip-flop vezérlése történhet statikus vezérlőjelet (T) követően a dinamikus bemeneten alkalmazott $1 \rightarrow 0$ ill. $0 \rightarrow 1$ átmenettel, vagy kizárólag a dinamikus bemenetre juttatott átmenettel. Az előbbi áramkört kétütemű, az utóbbit együtemű T flip-flopnak nevezzük.

T	Q	Q()	G_0	G_1	Flip flop állapota
0	0	0	0	0	Változatlan
0	1	1	0	0	
1	0	1	0	1	Ellentétes állapotba billen
1	1	0	1	0	



$$Q(\tau) = \bar{Q}T + Q\bar{T} = Q \oplus T$$

2-29. ábra

Kétütemű statikus kapuzású, dinamikus vezérlésű T flip-flop vezérlési táblázata és megvalósítása

a) Kétütemű T flip-flop igazságtáblázatát a kapuzó bemeneteken alkalmazandó vezérlésekkel 2-29. ábra mutatja. A statikus kapuzó bemenetek vezérlése:

$$G_0 = T \bar{Q};$$

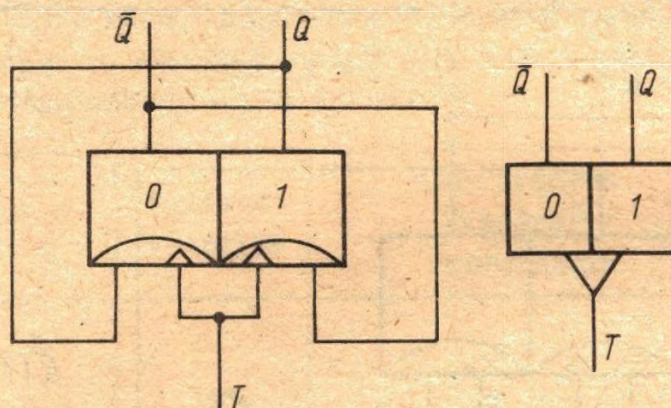
$$G_1 = T Q$$

Ellenőrzésképpen határozzuk meg a tároló logikai egyenletét. Mivel:

$$Q(\tau) = G_1 + \bar{G}_0 Q = TQ + (\bar{T} + \bar{Q})Q = TQ + \bar{T}Q = T \oplus Q$$

b) Együtemű T flip-flop a 2-29. ábra alapján közvetlenül előállítható a statikus T vezérlés elhagyásával. A tároló a V dinamikus bemenetre érkező vezérlőjel (órajel) hatására ellentétes állapotba billen (2-30. ábra). Az ábra megrajzolásakor $0 \rightarrow 1$ átmenetre működő tárolót tételtünk fel, vagyis az $1 \rightarrow 0$ átmenet a tárolóra nézve hatástalan.

T	Q	Q()	Flip flop állapota
$1 \rightarrow 0$	0	0	Változatlan
$1 \rightarrow 0$	1	1	
$0 \rightarrow 1$	0	1	Ellentétes állapotba billen
$0 \rightarrow 1$	1	0	



2-30. ábra

Együtemű, dinamikus vezérlésű tároló igazságtáblázata, megvalósítása, szimbolikus jelölése

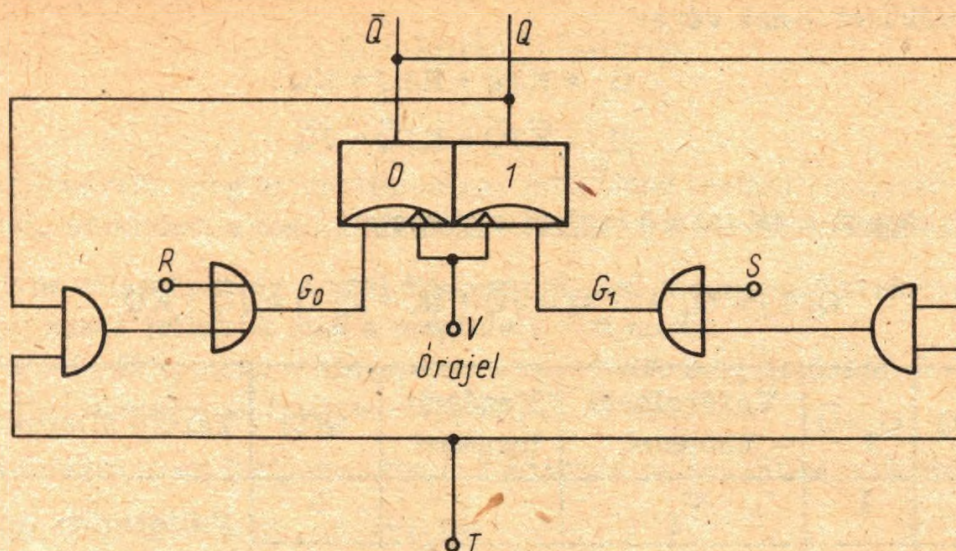
2.3.4.3. Dinamikus RST flip-flop

Könnyen belátható, hogy dinamikus vezérlésű RST flip-flop az $RS=RT=ST=0$ feltételek teljesülése esetén a 2-31. ábra szerint realizálható.

2.3.4.4. Dinamikus JK flip-flop

A T flip-flophoz hasonlóan a JK flip-flop is realizálható kétütemű és együtemű áramkörként.

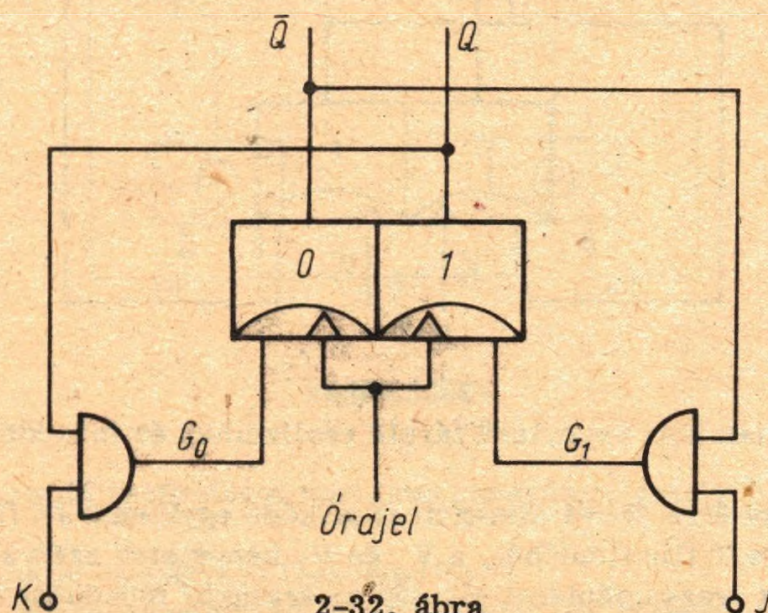
a) Kétütemű JK flip-flop igazságtáblázatát G_0 és G_1 vezérlésével kiegészítve előállítjuk a tároló vezérlési táblázatát (2-32. ábra). A vezérlési táblázat G_0 és G_1 igazságtáblázatának tekinthető a K, J, Q jeleket bemenő válto-



2-31. ábra

Statikus kapuzásu, dinamikus vezérlésű RST tároló megvalósítása

in-dex	K	J	Q	Q(τ)	G ₀	G ₁	Tároló állapota
0	0	0	0	0	x	0	Változatlan
1	0	0	1	1	0	x	Változatlan
2	0	1	0	1	0	1	1-es állapotba billen
3	0	1	1	1	0	x	1-es állapotban marad
4	1	0	0	0	x	0	0-ás állapotban marad
5	1	0	1	0	1	0	0-ás állapotba billen
6	1	1	0	1	0	1	Ellentétes állapotba billen
7	1	1	1	0	1	0	Ellentétes állapotba billen



2-32. ábra

Statikus kapuzásu, dinamikus vezérlésű JK flip-flop vezérlési táblázata és megvalósítása

zóknek tekintve. A szabadon választható vezérléseket az ábrán kereszttel je-
löltük. Értéküket 0-nak véve:

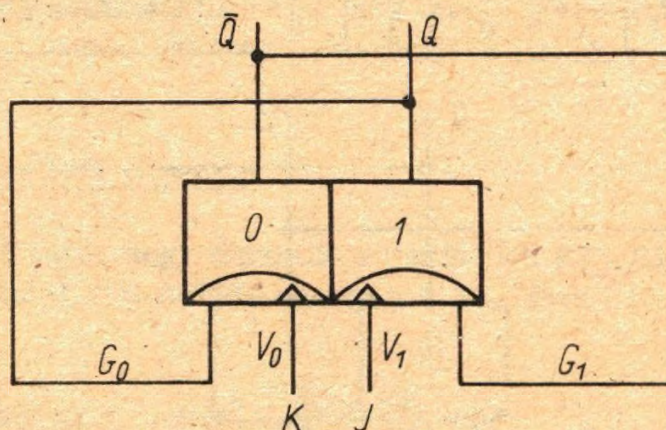
$$G_0 = K\bar{J}Q + KJQ = KQ$$

$$G_1 = \bar{K}J\bar{Q} + KJ\bar{Q} = J\bar{Q}$$

Ellenőrzésképpen a JK tároló logikai egyenlete:

$$Q(\tau) = G_1 + \bar{G}_0 Q = J\bar{Q} + (\bar{K} + \bar{Q}) Q = J\bar{Q} + \bar{K}Q$$

Q	$G_0 = Q$	$G_1 = \bar{Q}$	Vezérelhető bemenet	Vezérelt bemenet	$Q(\tau)$	Tároló állapota
0	0	1	V_1	-	0	Változatlan
1	1	0	V_0	-	1	
0	0	1	V_1	V_1	1	1-es állapotba billen
1	1	0	V_0	V_1	1	1-es állapotban marad
0	0	1	V_1	V_0	0	0-ás állapotban marad
1	1	0	V_0	V_0	0	0-ás állapotba billen
0	0	1	V_1	$V_0 = V_1$	1	Ellentétes állapotba billen
1	1	0	V_0	$V_0 = V_1$	0	



2-33. ábra

Együtemű, dinamikus vezérlésű tároló realizálása és működési táblázata

b) Negatív élre ($0 \rightarrow 1$ átmenetre) működő együtemű JK flip-flop készí-
thető a 2-30. ábra T flip-flopjából, a V_0 és V_1 bemenetek szétválasztásával
(2-33. ábra). Az igazságtáblázat alapján az áramkör működése könnyen ellen-
őrizhető.

2.4 Gyakorló feladatok

1. Tervezzük meg az un. impulzus működésű logikai rendszer tároló áramköreit, amelyben a logikai szintek leképzése a következőképpen történik:

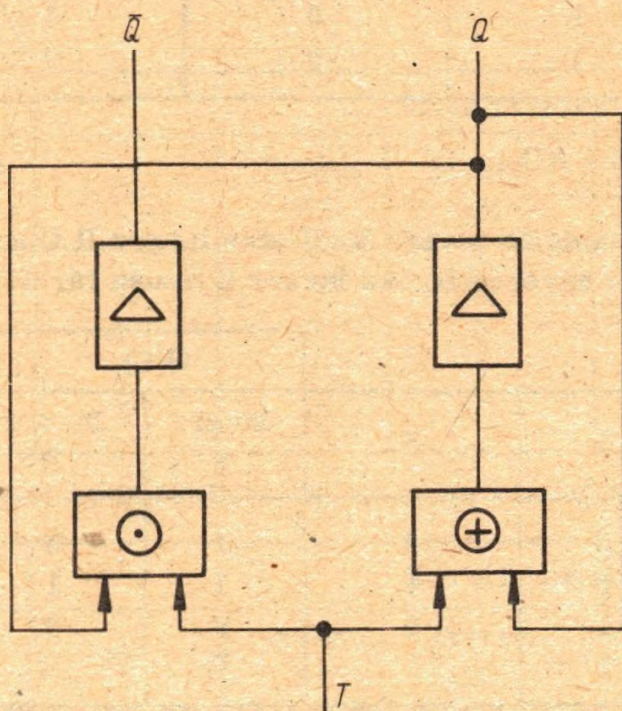
- a) 1-es állapotban az órajel frekvenciájával megegyező frekvenciájú impulzusok jelennek meg a megfelelő kimeneten.
- b) "0"-ás állapotban a kimeneten nem jelenik meg az órajel frekvenciájával megegyező frekvenciájú impulzus sorozat.

Határozzuk meg D típusú tárolót felhasználva impulzus működésű

- a) RS, JK, inverz RS, inverz JK
- b) T, RST, inverz T

típusú tároló logikai hatásvázlatát. A tervezéshez használjuk fel a tárolók logikai egyenletét.

(Pl. T típusú impulzus működésű tároló egyszerűsített logikai hatásvázlatát 2-34. ábra szemlélteti.)



2-34. ábra

2. Tervezendő vezérelhető impulzus generátor, amely V vezérlőjel hatására

- a) az órajellel megegyező frekvenciájú
- b) az órajel frekvenciájának felével megegyező frekvenciájú impulzus sorozatot állít elő.

3. Határozzuk meg az "S" típusú tároló állapot diagramját és logikai egyenletét. Az S flip-flop vezérlési táblázata:

R	S	$Q(\tau)$
0	0	Q
0	1	1
1	0	0
1	1	1

Megoldás: $Q(\tau) = S + \bar{R}Q$.

4. Határozzuk meg az "R" típusú tároló állapot diagramját és logikai egyenletét. Az R típusú flip-flop vezérlési táblázata:

R	S	$Q(\tau)$
0	0	Q
0	1	1
1	0	0
1	1	0

Megoldás: $Q(\tau) = \bar{R}S + \bar{R}Q$.

5. Határozzuk meg az inverz S, illetve inverz R típusú tároló állapot diagramját és logikai egyenletét. Az inverz S típusú tároló vezérlési táblázata:

R	S	$Q(\tau)$	
		Inverz S	Inverz R
0	0	1	0
0	1	1	1
1	0	0	0
1	1	Q	Q

Megoldás:

inverz S: $Q(\tau) = \bar{R} + SQ$

inverz R: $Q(\tau) = \bar{R}S + SQ$.

6. Határozzuk meg az "E" típusú tároló állapot diagramját és logikai egyenletét. Az E típusú tároló vezérlési táblázata:

R	S	Q(T)
0	0	Q
0	1	1
1	0	0
1	1	Q

Megoldás: $Q(T) = \overline{R}S + SQ + \overline{R}Q = \overline{R}(S+Q) + SQ = \overline{R}(S+Q) + SQ = R + S+Q + SQ = R + S + Q + \overline{S} + \overline{Q} = R + S + Q$

7. Határozzuk meg a következő tárolók kapcsolási rajzát NEMVAGY műveletet megvalósító áramköri elemek felhasználásával:

- S, illetve R típusú flip-flop
- Inverz S, illetve inverz R típusú flip-flop
- E típusú flip-flop.

8. Az irodalomban használatos az inverz működésű tárolók alábbi értelmezése is:

Bemenet		Tároló típusa				
R ill. K	S ill. J	RS	Inverz RS	Inverz JK	Inverz S	Inverz R
0	0	Q	?	$\overline{Q}$	1	0
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	1	?	Q	Q	Q	Q

Összehasonlításképpen feltüntetjük az eddig használt értelmezést is:

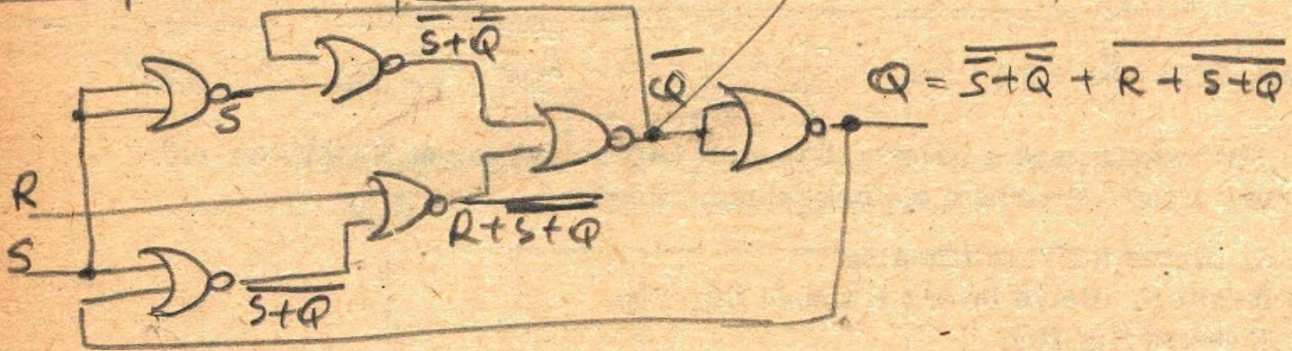
Bemenet		Tároló típusa				
R ill. K	S ill. J	RS	Inverz RS	Inverz JK	Inverz S	Inverz R
0	0	Q	?	$\overline{Q}$	1	0
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	1	?	Q	Q	Q	Q

Határozzuk meg a bemutatott értelmezés szerint is a tárolók állapot diagramját, állapot táblázatát, logikai egyenletét.

9. Valósítsuk meg

- RS tárolót D típusú flip-flop felhasználásával.
- RST és JK típusú tárolót D típusú flip-flop felhasználásával.
- T típusú tárolót JK típusú tároló felhasználásával.
- JK típusú tárolót T típusú tároló felhasználásával.

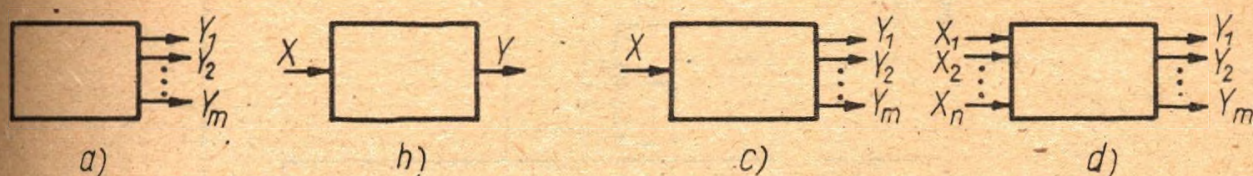
H/c. példák megoldása



3. SZINKRON MŰKÖDÉSŰ SZEKVENCIÁLIS HÁLÓZATOK

3.1 Szekvenciális hálózatok felosztása

A szekvenciális hálózatokat a bemenő ill. kimenő jelek száma alapján feloszthatjuk szekvenciális generátorokra, szekvenciális átalakítókra, számláncokra és általános szekvenciális hálózatokra (3-1. ábra).



3-1. ábra

Szekvenciális hálózatok

- (a) szekvenciális generátor, (b) szekvenciális átalakító,
(c) számlánc, (d) általános szekvenciális hálózat

a) Szekvenciális generátor (vagy autonom hálózat) az órajelek ütemében periodikusan bináris jelsorozatot állít elő, melyet a hálózat kimenő kapcsairól vehetünk le. A szekvenciális generátornak külön vezérlő bemenő jele nincs. A szinkronjelekre működő hálózat egyéb elnevezése szinkron szekvenciális automata (gép).

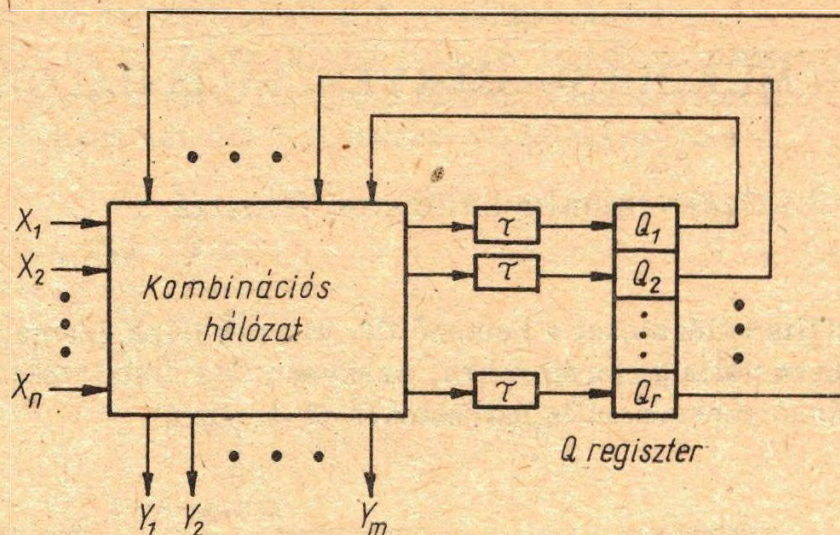
b) Szekvenciális átalakító egy bemenettel és egy kimenettel rendelkezik. Periodikusan ismétlődő bemenő jelekre a bemenetével azonos periodicitású periodikusan ismétlődő kimenő jelet ad, a kimenő jelsorozat függ a bemenő jelsorozattól és a hálózat felépítésétől.

c) Számláncok a legismertebb szekvenciális hálózatok. A bemenetre érkező jelek számlálása történhet bináris, decimális stb. kijelzéssel. A számláncok a feltüntetett bemeneten kívül rendszerint egy törlő bemenettel is rendelkeznek.

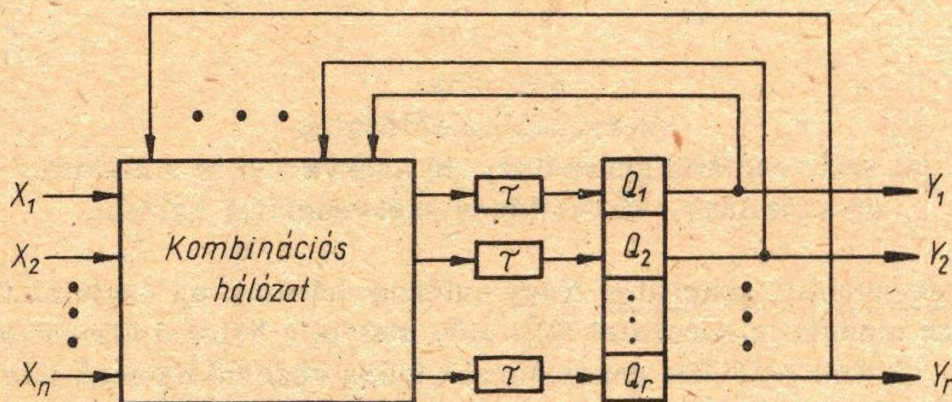
d) Általános szekvenciális hálózat több bemenettel és több kimenettel rendelkezik. Az előzőleg tárgyalt hálózatok az általános szekvenciális hálózat speciális esetei. A szekvenciális hálózatokat két csoportba oszthatjuk, ezeket A és B típusu hálózatoknak nevezik.

Az általános A típusu szekvenciális hálózat (3-2. ábra) $(n+r)$ bemenetű, m kimenetű kombinációs hálózatból és r tárolóból (cellából) álló regiszterből

épül fel. Az ábrán az áramköri késleltetéseket külön feltüntettük. A regiszter bármilyen típusú tárolóval megvalósítható.



a)



b)

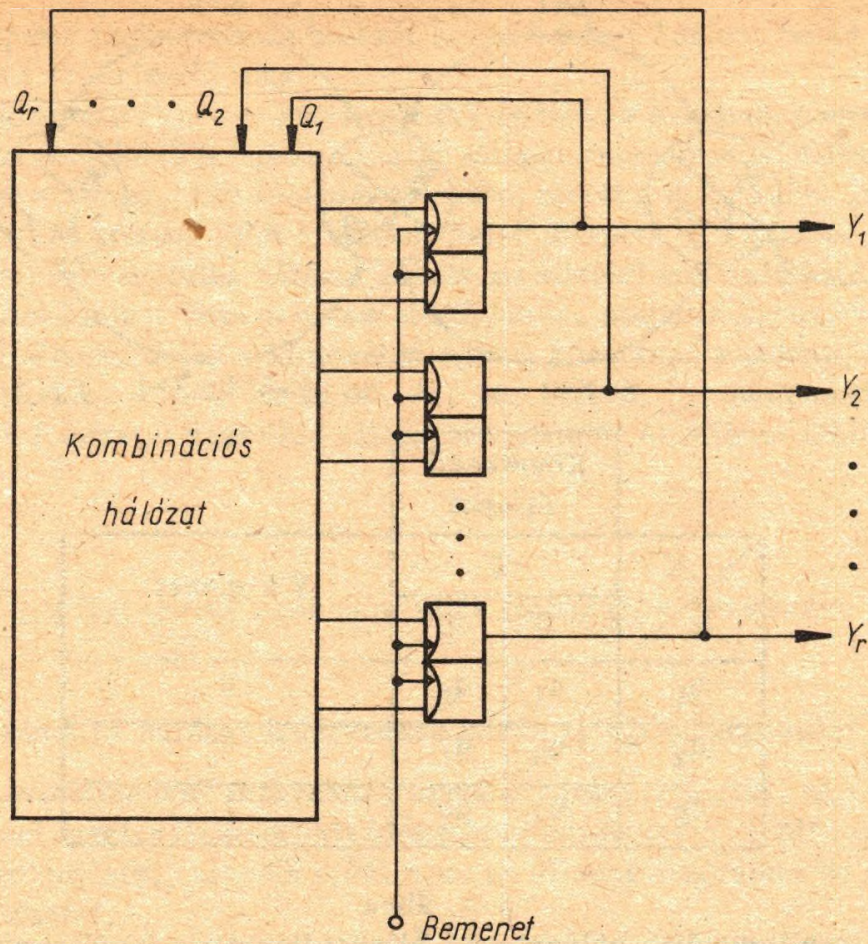
3-2. ábra

Általános szekvenciális hálózatok

(a) A típusu, (b) B típusu szekvenciális hálózat

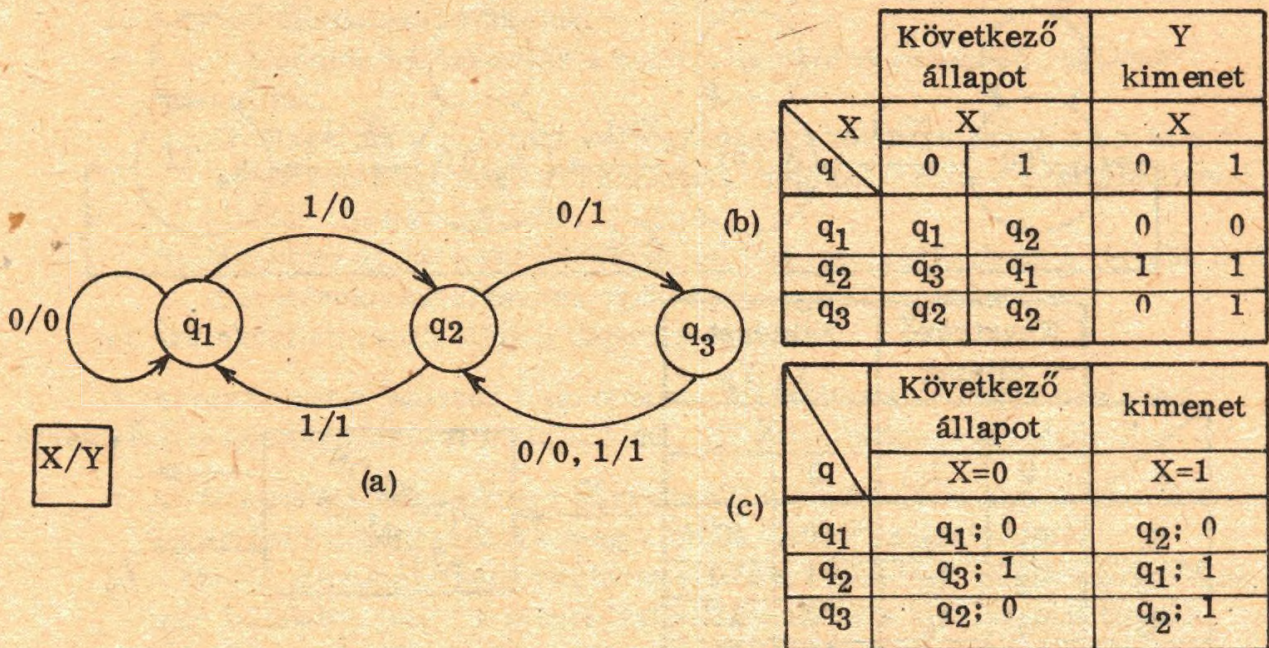
A B típusu szekvenciális hálózat az A típus speciális esetének is tekinthető. A kimenetek száma kisebb is lehet mint a tároló elemek száma. Jól ismert B típusu szekvenciális hálózat a szinkron számlánc (3-3. ábra).

Egy bemenetű (X) egy kimenetű (Y) három szekunder változóval (q_1, q_2, q_3) rendelkező A típusu szekvenciális hálózat állapot diagramját és állapot táblázatát láthatjuk a 3-4. ábrán. Az állapot táblázat tartalmazza a bemenő jel hatására létrejövő következő belső állapotot (q) és a kimenő jelet. Ezt a két adatot vagy a táblázat két külön oszlopában, vagy egymás mellett közös oszlopban adhatjuk meg. Pl. q_1 belső állapotból indulva 1, 0, 0, 1 bemenő jelsorozat hatására a kimenő jelsorozat: 0, 1, 0, 1 és a bemenő vezérlés hatására visszatérünk a kiinduló belső állapothoz.



3-3. ábra

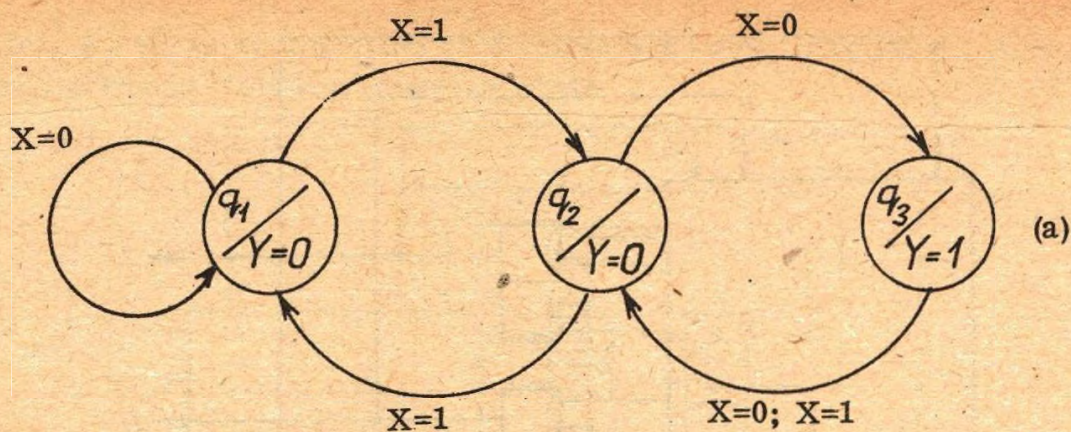
Szinkron számlánc, mint B típusú kombinációs hálózat



3-4. ábra

A típusú szekvenciális hálózat

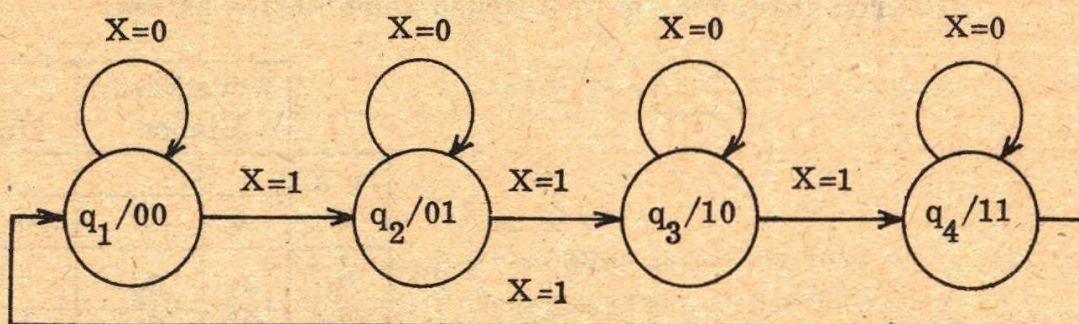
(a) Állapotdiagram, (b) és (c) állapot táblázat különböző alakjai



		Következő állapot		
q \ X	X	X		Y kimenet
		0	1	
q ₁		q ₁	q ₂	0
q ₂		q ₃	q ₁	0
q ₃		q ₂	q ₂	1

3-5. ábra

B típusú szekvenciális hálózat. (a) Állapotdiagram, (b) állapottáblázat



		Következő állapot		Kimenet Y ₁ Y ₂	
q \ X	X	X		X	
		0	1	0	1
q ₁		q ₁	q ₂	00	01
q ₂		q ₂	q ₃	01	10
q ₃		q ₃	q ₄	10	11
q ₄		q ₄	q ₁	11	00



3-6. ábra

Szinkron számlánc állapot diagramja és táblázata

"A" típusu szekvenciális hálózat kimenő jele a hálózat belső állapotától és a bemeneti vezérléstől függ.

Egy bemenetű (X), egy kimenetű (Y) és három szekunder változóval (q_1, q_2, q_3) rendelkező B típusu hálózat állapot diagramja és állapot táblázata hasonlóan adható meg. Egy lehetséges esetet mutat a 3-5. ábra. B típusu szekvenciális hálózat kimenő jelét a hálózat belső állapota (q_1) határozza meg.

Pl. Szinkron szekvenciális hálózat esetén ez közvetlenül belátható, mert a bemenő jel változásainak hatása a késleltető hatású statikus kapuzó bemeneteken azonnal megjelenik, de ez a kimeneten csak a következő szinkron (óra) jel hatására jelentkezik. A 3-6. ábrán 00, 01, 10, 11, 00, ... sorrendben számláló két bites (két fokozatu) számláló állapot diagramját és állapot táblázatát tüntetjük fel.

3.2 Szinkron szekvenciális hálózatok tervezése

Szekvenciális hálózatok tervezése során az alábbi lépéseket követhetjük.

- a) A követelmények (szavakban megadott leírás, logikai algebra módszereivel megfogalmazott kívánságok, ütemdiagram alapján előírt tulajdonságok stb.) alapján előállítjuk az állapot diagramot és az állapot táblázatot.
- b) Az állapot táblázatot egyszerűsítjük, meghatározzuk a feladat megoldásához minimálisan szükséges szekunder változók (belső állapotok) számát.
- c) A belső állapotokat kódoljuk.
- d) A kombinációs hálózatot és a belső állapotokat realizáló regisztert realizáljuk.
 1. Kiválasztjuk a műszaki vagy gazdasági követelmények alapján a legkedvezőbb áramköri készletet. Meghatározzuk a felhasználandó tárolók típusát (RS, JK, T stb.).
 2. Meghatározzuk a kombinációs hálózat kimenetének és a tárolók bemenetének vezérléséhez szükséges logikai egyenleteket.
- e) Áramkörileg realizáljuk a kapott megoldást.

A tervezés során először a tárolók vezérléséhez szükséges bemeneti egyenletek meghatározásával foglalkozunk. Ez kétféleképpen történhet.

- a) Az állapot táblázat alapján közvetlenül a kiválasztott típusu tároló elem vezérlési tulajdonságának megfelelően (közvetlen módszer).
- b) A tároló elemmel szembeni követelmények alapján, általánosan oldjuk meg feladatot, majd a megoldást alkalmazzuk a kiválasztott típusu tároló elemre (általános módszer).

3.2.1. A tárolók vezérléséhez szükséges bemeneti egyenletek közvetlen meghatározása

A közvetlen tervezés bemutatására tervezzük meg egy aritmetikai egység akkumulátor regiszterét, amely a következő tulajdonságokkal rendelkezik:

- a) V_+ összeadást vezérlő jel hatására az A és B regiszter tartalmát helyértékenként összeadja, az eredményt valamennyi helyértéken módosítja az előző (alacsonyabb) helyértéken keletkező átvittel. A végleges összeget az A regiszterben tárolja.
- b) V_- kivonást vezérlő jel hatására az A és B regiszter tartalmát kivonja egymásból, a különbségből kivonja az előző (alacsonyabb) helyértéken előállított áthozatot (átvételt). A két kivonás eredményét az A regiszterben tárolja.
- c) V_J jel hatására az A regiszter tartalmát egy helyértékkal jobbra lépteti.
- d) V_B jel hatására az A regiszter tartalmát egy helyértékkal balra lépteti.
- e) V_T törlő jel hatására az akkumulátor regiszter minden tárolóját törli.

A megoldáshoz kétütemű, statikus kapuzású, dinamikus vezérlésű T tárolót használunk. Határozzuk meg az A regiszter i -edik cellájának (elemének) vezérlési feltételeit.

Az A ill. B regiszter i -edik celláját képező tároló kimenetét A_i ill. B_i -vel az előző alacsonyabb helyértékről érkező átvitelt C_{i-1} , átvitelt R_{i-1} -gyel jelölve a feladat szimbolikus jelöléssel felírva:

$$V_+ : (A_i) + (B_i) + (C_{i-1}) \rightarrow A_i; \quad V_- : (A_i) - (B_i) - (R_{i-1}) \rightarrow A_i;$$

$$V_J : (A_{i-1}) \rightarrow A_i; \quad V_B : (A_{i+1}) \rightarrow A_i; \quad V_T : 0 \rightarrow A$$

Az összeadás és kivonás igazságtáblázata (3-7. ábra) (független változóként) tartalmazza az A_i , B_i , C_{i-1} ill. R_{i-1} k -edik ütemben felvett értékeinek kombinációit $(A_i^k, B_i^k, C_{i-1}^k, R_{i-1}^k)$, valamint az eredményt a $(k+1)$ -edik ütemben (A_i^{k+1}) .

Közvetlen tervezés során a tároló bemenetének vezérlési feltételét a tároló igazságtáblázata alapján állapítjuk meg. T típusú tároló esetén T_A érték az igazságtáblázat azon helyein 1, ahol A_i^k és A_i^{k+1} különböző. A Veitch diagramon elvégzett egyszerűsítés után összeadás ill. kivonás esetén a tároló bemenetének vezérlési feltétele:

$$T_{Ai} = V_+ (B_i^k \oplus C_{i-1}^k);$$

in- dex	k-adik ütem			(k+1)- edik ütem	beme- net ve- zérlése	Megjegyzés
	A_i^k	B_i^k	C_{i-1}^k R_{i-1}^k	A_i^{k+1}	T_{Ai}	
0	0	0	0	0	0	
1	0	0	1	1	1	Tároló ellentétes állapotba billen
2	0	1	0	1	1	Tároló ellentétes állapotba billen
3	0	1	1	0	0	
4	1	0	0	1	0	
5	1	0	1	0	1	Tároló ellentétes állapotba billen
6	1	1	0	0	1	Tároló ellentétes állapotba billen
7	1	1	1	1	0	

(a)

		B_i^k	0	0	1	1
		C_{i-1}^k	0	1	1	0
A_i^k						
			0	1	3	2
0			1		1	
1			4	5	7	6
			1		1	

(b)

3-7. ábra

Összeadás ill. kivonás vezérlési táblázata és Veitch diagramja

ill.

$$T_{Ai} = V_- (B_i^k \oplus R_{i-1}^k)$$

Jobbra ill. balra léptetés igazságtáblázatát (3-8. ábra) kiolvasva:

$$T_{Ai} = V_J (A_{i-1}^k \oplus A_i^k),$$

ill.

$$T_{Ai} = V_B (A_{i+1} \oplus A_i^k)$$

Az akkumulátor regiszter törlésének feltétele:

$$T_{Ai} = V A_i$$

A_{i-1}^k	A_i^k	A_i^{k+1}	T_{Ai}
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

(a)

A_{i+1}^k	A_i^k	A_i^{k+1}	T_{Ai}
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

(b)

A_i^k	A_i^{k+1}	T_{Ai}
0	0	0
1	0	1

(c)

3-8. ábra

(a) Léptetés jobbra, (b) Léptetés balra, (c) Törlés vezérlési táblázata

Az akkumulátor regiszter logikai hatásvázlatát a 3-9. ábra mutatja. Az ábrán az egyszerűség kedvéért az antivalencia kapukat csak szimbolikusan rajzoltuk fel.

A feladatot oldjuk most meg dinamikus vezérlésű, statikus kapuzású RS tárolóval. A 3-10. ábra igazságtáblázatán a szabadon választható vezérléseket kereszttel jelöljük. Kapuzó bemenetek vezérlésének biztosítania kell

$A_i^k \rightarrow A_i^{k+1}$ változást, $G_0 \cdot G_1 = 0$ feltétel teljesítésével. A bemenetek vezérlése pl. összeadás esetén:

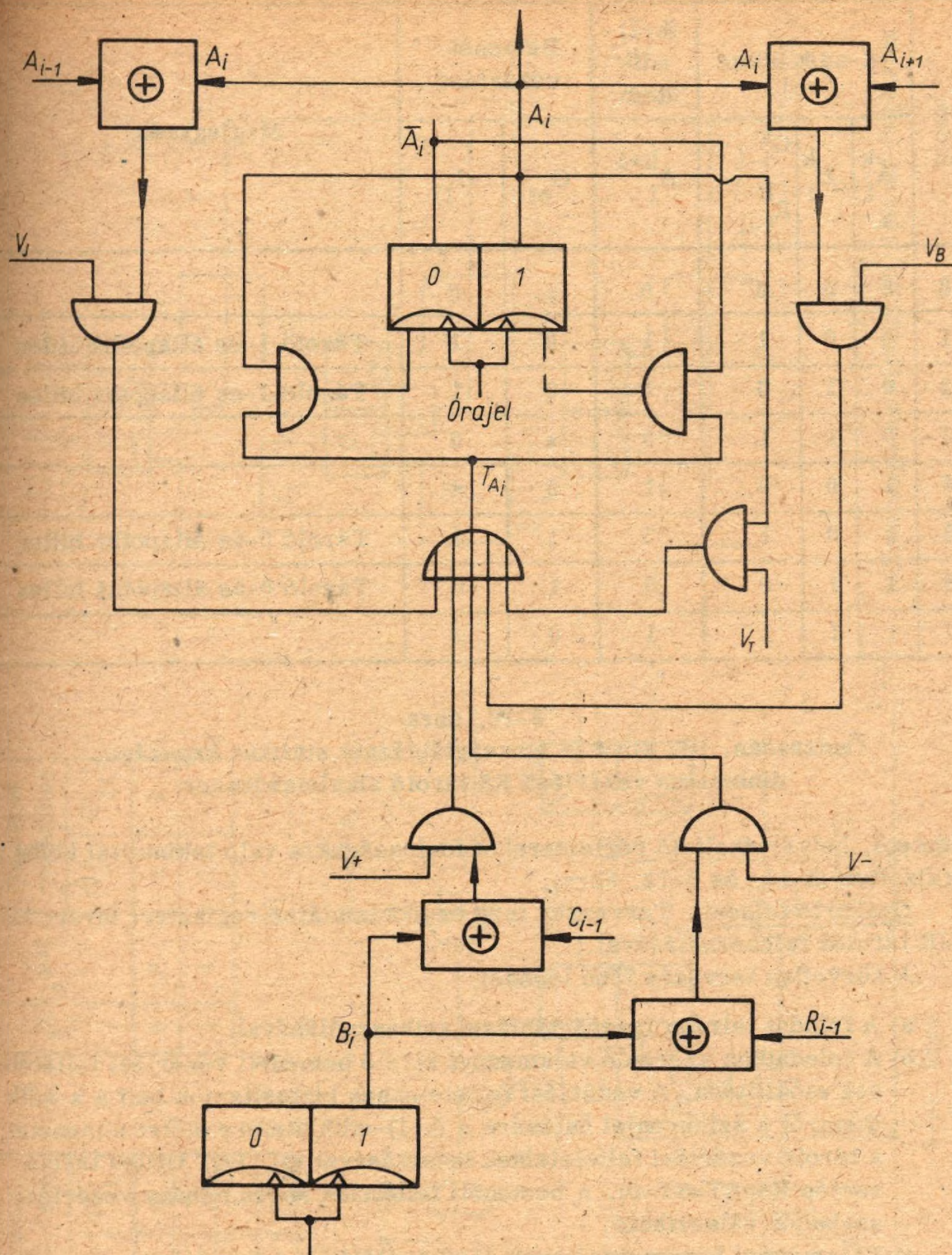
$$G_{oi} = V_+ (m_5 + m_6) = V_+ A_i^k (\overline{B_i^k} C_{i-1}^k + B_i^k \overline{C_{i-1}^k})$$

$$G_{li} = V_+ (m_1 + m_2) = V_+ \overline{A_i^k} (\overline{B_i^k} C_{i-1}^k + B_i^k \overline{C_{i-1}^k})$$

A jobbra léptetés igazságtáblázatát (3-11. ábra) felhasználva

$$G_{oi} = V_J (\overline{A_{i-1}^k} \overline{A_i^k} + \overline{A_{i-1}^k} A_i^k) = V_J \overline{A_{i-1}^k}$$

$$G_{li} = V_J (A_{i-1}^k \overline{A_i^k} + A_{i-1}^k A_i^k) = V_J A_{i-1}^k$$



3-9. ábra
Akkumulátor regiszter egyszerűsített logikai hatásvázlata

Index	k-adik ütem			(k+1)- edik ütem	Bemenet vezérlése		Megjegyzés
	A_i^k	B_i^k	C_{i-1}^k R_{i-1}^k	A_i^{k+1}	G_{oi}	G_{li}	
0	0	0	0	0	x	0	
1	0	0	1	1	0	1	Tároló 1-es állapotba billen
2	0	1	0	1	0	1	Tároló 1-es állapotba billen
3	0	1	1	0	x	0	
4	1	0	0	1	0	x	
5	1	0	1	0	1	0	Tároló 0-ás állapotba billen
6	1	1	0	0	1	0	Tároló 0-ás állapotba billen
7	1	1	1	1	0	x	

3-10. ábra

Összeadás, ill. kivonás igazságtáblázata statikus kapuzásu, dinamikus vezérlésű RS tároló alkalmazásakor

A léptető, helyértékeltoló regisztereket fontosságukra való tekintettel külön is felrajzoltuk (3-11. és 3-12. ábra).

Gyakorló feladat. Tervezzük meg az akkumulátor regisztert Inverz RS és JK tárolók felhasználásával.

A közvetlen tervezés főbb lépései:

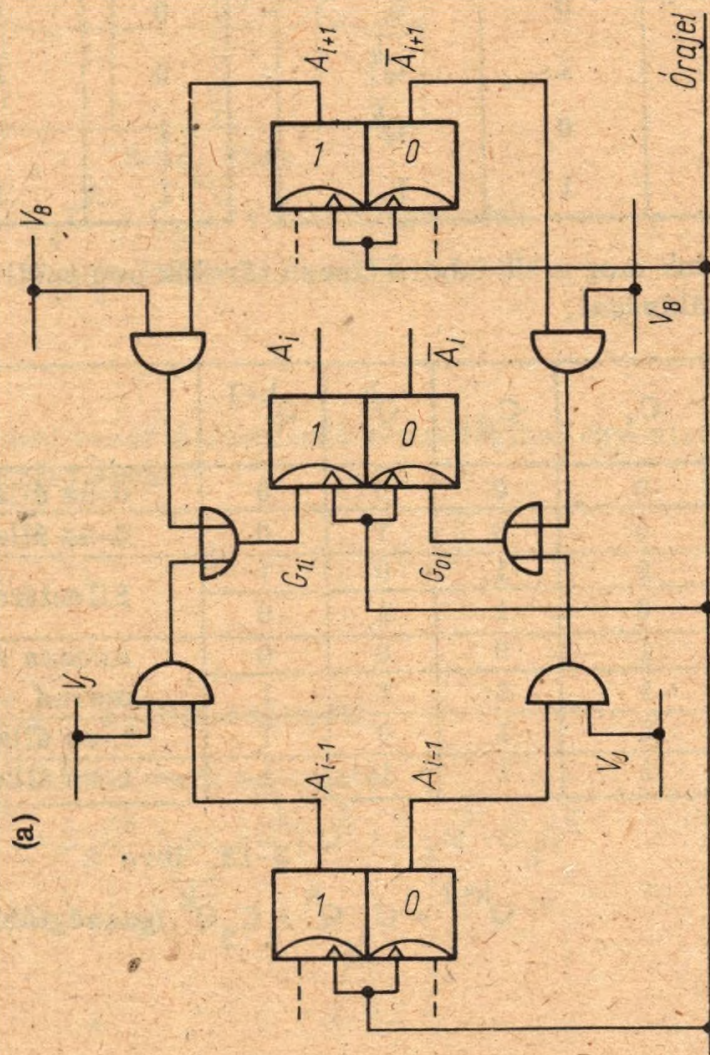
- A feladat leírása igazságtáblázat felhasználásával.
- A feladatban szereplő valamennyi tároló bemeneti vezérlési feltételeinek előállítása. A vezérlési feltételeknek biztosítaniuk kell a k-adik ütemről a szinkronjel hatására a (k+1)-edik ütemre előírt átmenetet, a tároló vezérlési feltételeinek betartásával (pl. RST típusú tároló esetén $RS=RT=ST=0$). A bemeneti feltételek során néhány vezérlés szabadon választható.
- Valamennyi bemenetre kapott logikai feltételt (egyenletet) minimalizáljuk (pl. Veitch diagram alapján).
- A kapott megoldást realizáljuk.
- Más típusú tároló választása esetén a tervezés b, c, d pontjait megismételjük.

A_{i-1}^k	A_i^k	A_i^{k+1}	Bemenet vezérlése	
			G_{oi}	G_{li}
0	0	0	x	0
0	1	0	1	0
1	0	1	0	1
1	1	1	0	x

A_i^k	A_i^{k+1}	Bemenet vezérlése	
		G_{oi}	G_{li}
0	0	x	0
1	0	1	0

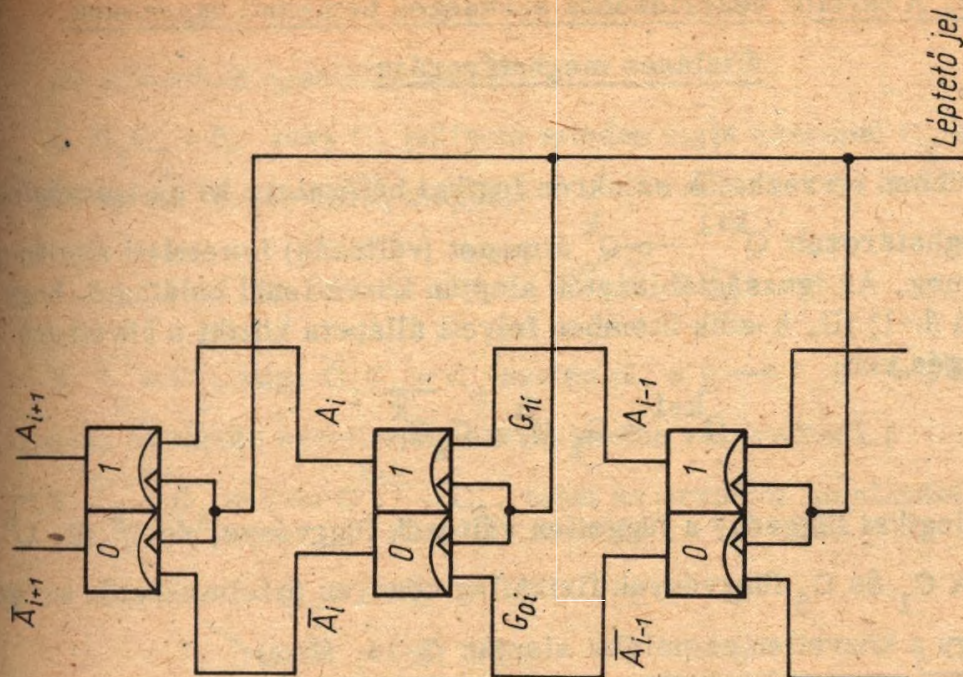
(b)

(a)



3-11. ábra

- (a) Léptetés jobbra, (b) Törlés igazságtáblázata,
(c) Kétirányu helyértéktoló áramkör logikai vázlata



3-12. ábra

Egyirányu helyértéktoló, léptető regiszter

3.2.2. A tárolók vezérléséhez szükséges bemeneti egyenletek

általános meghatározása

Egyszerűbben tervezhetők szinkron logikai hálózatok, ha az igazságtáblázat alapján meghatározott $Q^{k+1} \rightarrow Q^k$ átmenet (változás) feltételeit általános esetre odjuk meg. Az igazságtáblázatok alapján közvetlenül belátható, hogy bármely tároló $(k+1)$ ill. k -adik ütemben felvett állapota között a következő alaku összefüggés van:

$$Q^{k+1} = C_1 Q^k + C_2 \overline{Q^k}$$

ahol C_1 és C_2 logikai függvény a független változók függvénye, de Q^k ill. $\overline{Q^k}$ -től független. A C_1 és C_2 függvények fizikailag könnyen értelmezhetők az igazságtáblázat vagy a közvetlen szemlélet alapján (3-13. ábra):

C_1	C_2	Q^{k+1}	Q^k	Q^{k+1}	C_1, C_2 és Q^k kombinációja
0	0	0	0	0	$\overline{C_2} \overline{Q^k}$
0	1	$\overline{Q^k}$	0	1	$C_2 \overline{Q^k}$
1	0	Q^k	1	0	$\overline{C_1} Q^k$
1	1	1	1	1	$C_1 Q^k$

Határozzuk meg a különböző típusú tárolók bemeneti egyenleteit C_1, C_2, Q^k felhasználásával.

index	C_1	C_2	Q^k	Q^{k+1}	Megjegyzés
0	0	0	0	0	0-ás állapotban marad
1	0	0	1	0	0-ás állapotba billen
2	0	1	0	1	Ellentétes állapotba billen
3	0	1	1	0	
4	1	0	0	0	Azonos állapotban marad
5	1	0	1	1	
6	1	1	0	1	1-es állapotba billen
7	1	1	1	1	1-es állapotban marad

3-13. ábra

$$Q^{k+1} = C_1 Q^k + C_2 \overline{Q^k} \text{ igazságtáblázata}$$

Az előforduló speciális esetek:

1. $\overline{C_1}C_2 = 0$, azaz C_1 polinom minden tagja szerepel C_2 -ben, akkor a Veitch diagram (3-14. ábra) alapján:

$$Q^{k+1} = C_1 Q^k + C_2$$

2. $\overline{C_1} = C_2$, vagy $\overline{C_1}C_2 = C_2$ esetén pl. a $0 \rightarrow 1$, $1 \rightarrow 0$ billentés feltétele azonos. (A tároló egyetlen bemeneti ponton vezérelhető.)

3. $C_1 = C_2$ esetén $Q^{k+1} = C_1$, tehát az áramkör kombinációs hálózat.

		C_2			
		0 0 1 1			
		Q^k			
		0 1 1 0			
C_1					
	0			x	1
	1		1	1	1

$$Q^{k+1} = C_1 Q^k + C_2$$

3-14. ábra

$$Q^{k+1} = C_1 Q^k + C_2 \overline{Q^k} \text{ Veitch diagramja}$$

$$\overline{C_1}C_2 = 0 \text{ feltétel esetén}$$

3.2.2.1. RS flip-flop

Az RS flip-flop bemeneti vezérléséhez szükséges logikai egyenletek dinamikus vezérlésű tárolót feltételezve:

$$G_0 = \overline{C_1} Q^k \quad \text{ill.} \quad G_1 = C_2 \overline{Q^k}$$

Ellenőrzésképpen:

a) $G_0 \cdot G_1 = \overline{C_1} C_2 Q^k \overline{Q^k} = 0$ feltétel teljesül

b) Az RS tároló logikai differencia egyenlete:

$$Q^{k+1} = G_1 + \overline{G_0} Q^k = C_2 \overline{Q^k} + (\overline{C_1} + Q^k) Q^k = C_1 Q^k + C_2 \overline{Q^k}$$

a kiindulási állapot egyenletet adja.

Abban a speciális esetben, amikor $\overline{C_1}C_2 = 0$ pl. az állapot egyenletek együtthatóinak összehasonlításából:

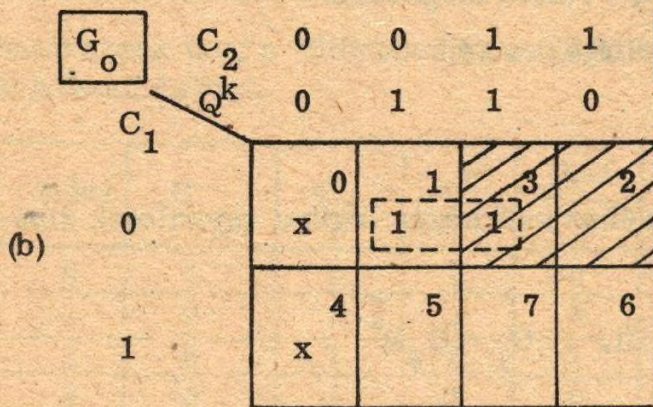
$$Q^{k+1} = C_1 Q^k + C_2 \quad \text{ill.} \quad Q^{k+1} = G_1 + \overline{G_0} Q^k$$

$$G_0 = \overline{C_1} \quad \text{ill.} \quad G_1 = C_2$$

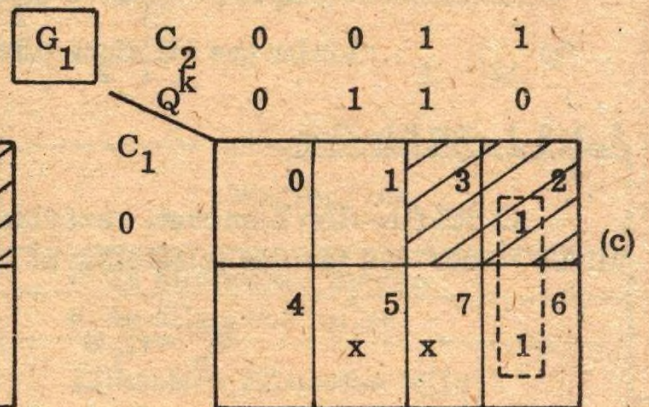
A bemeneti egyenletek meghatározását C_1, C_2, Q^k ismeretében nemcsak C_1, C_2 fizikai értelmezése, hanem az előző pontban bemutatott közvetlen módszer alapján is elvégezhetjük. A 3-15. ábra igazságtáblázatán a tetszőlegesen

(a)

index	C_1	C_2	Q^k	Q^{k+1}	G_0	G_1	K	J
0	0	0	0	0	x	0	x	0
1	0	0	1	0	1	0	1	x
2	0	1	0	1	0	1	x	1
3	0	1	1	0	1	0	1	x
4	1	0	0	0	x	0	x	0
5	1	0	1	1	0	x	0	x
6	1	1	0	1	0	1	x	1
7	1	1	1	1	0	x	0	x



$$G_0 = \overline{C_1} Q^k$$



$$G_1 = C_2 \overline{Q}^k$$

3-15. ábra

(a) RS és JK tároló vezérlési táblázata, (b) G_0 , (c) G_1 Veitch diagramja

megválasztható vezérléseket ismét kereszttel jelölve, G_0 ill. G_1 -re felrajzolt Veitch diagram (minterm tábla) alapján:

$$G_0 = \overline{C_1} Q^k,$$

$$G_1 = C_2 \overline{Q}^k$$

$\overline{C}_1 C_2 = 0$ speciális esetben a Veitch diagram vonalkázott termjeinek együtt-
hatóit is szabadon választhatjuk meg. Az egyes mintermek együttthatóit f_i -vel
jelölve,

$$f_0 = f_2 = 1, \quad f_4 = 0 \quad \text{választással: } G_0 = \overline{C}_1$$

$$f_3 = f_7 = 1, \quad f_5 = 0 \quad \text{választással: } G_1 = C_2$$

Ellenőrzésképpen: $G_0 G_1 = \overline{C}_1 C_2$ feltételünk értelmében zérus.

3.2.2.2. JK flip-flop

JK flip-flop bemeneteinek vezérlését végző logikai egyenleteket több uton
is meghatározhatjuk.

a) Közvetlen módszer alkalmazása. A vezérlési táblázat (3-15. ábra)
független változói: C_1, C_2, Q^k, J és K vezérlése $Q^k \rightarrow Q^{k+1}$ átmenetet kell,
hogy eredményezze.

A vezérlési táblázatban $Q^k \rightarrow Q^{k+1}$ átmenetet vizsgálva a következő ese-
tek fordulnak elő:

Q^k	Q^{k+1}	K	J	C_1, C_2 és Q^k kombinációja
0	0	X	0	$\overline{C}_2 \overline{Q^k}$
0	1	X	1	$C_2 \overline{Q^k}$
1	0	1	X	$\overline{C}_1 Q^k$
1	1	0	X	$C_1 Q^k$

A K ill. J bemenet Veitch diagramján (3-16. ábra) a határozatlan mintermek
együttthatóit pl. a következőképpen válasszuk meg:

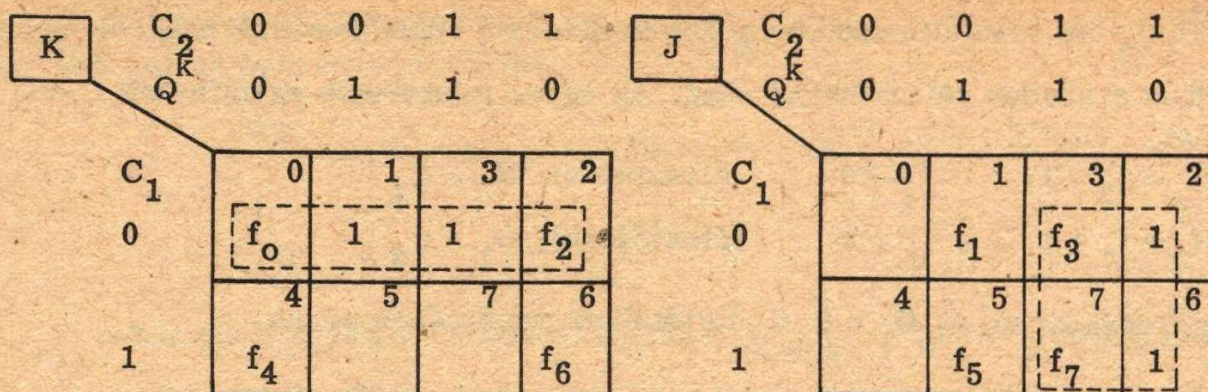
$$f_0 = f_2 = f_3 = f_7 = 1$$

és

$$f_1 = f_4 = f_5 = f_6 = 0$$

Ebben az esetben:

$$J = C_2 \quad \text{és} \quad K = \overline{C}_1$$



3-16. ábra
K és J bemenete vezérlési feltételeinek meghatározása
Veitch diagram alapján

Ha valamennyi határozatlan minterm együttthatóját 1-nek választjuk, akkor

$$J = C_2 \overline{Q^k} + Q^k \quad \text{ahol } Q^k \text{ a redundáns részekből származik}$$

$$K = \overline{C_1} Q^k + Q^k \quad \text{ahol } \overline{Q^k} \text{ a redundáns részekből származik.}$$

b) valamennyi redundáns részt figyelembe véve a közvetlen fizikai szemlélet alapján is:

$$J = C_2 \overline{Q^k} + Q^k \quad \text{ill.} \quad K = \overline{C_1} Q^k + \overline{Q^k}$$

Ezeket az egyenleteket a $Q^k \rightarrow Q^{k+1}$ átmenetre felírt táblázatból olvashatjuk ki.

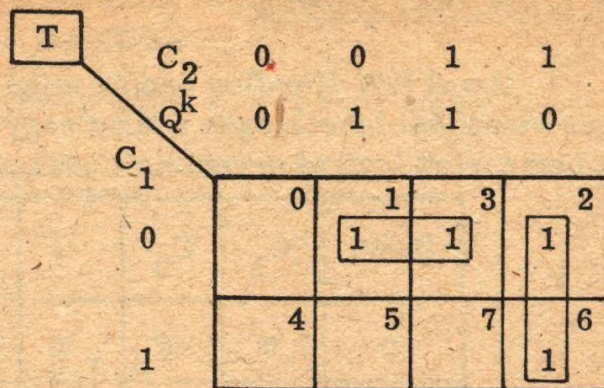
3.2.2.3. T flip-flop

T flip-flop bemenetét vezérlő logikai egyenletet a JK flip-flophoz hasonló módon határozzuk meg.

a) Közvetlen módszer alkalmazása. A vezérlési táblázatban (3-17. ábra) $Q^k \rightarrow Q^{k+1}$, $0 \rightarrow 1$, $1 \rightarrow 0$ átmenetet kell csak figyelembe vennünk, a másik két esetben a T bemenetet nem vezéreljük. Vagyis:

Q^k	Q^{k+1}	T	C_1, C_2, Q^k kombinációja
0	0	0	$C_2 \overline{Q^k}$ $\overline{C_1} Q^k$
0	1	1	
1	0	1	
1	1	0	

index	C_1	C_2	Q^k	Q^{k+1}	T
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	1
7	1	1	1	1	0



$$T = \overline{C_1} Q^k + C_2 \overline{Q^k}$$

$$\overline{T} = C_1 Q^k + \overline{C_2} \overline{Q^k}$$

3-17. ábra

T tároló vezérlési táblázata, bemenete vezérléséhez szükséges logikai egyenlet meghatározása Veitch diagram segítségével

A Veitch diagram alapján

$$T = \overline{C_1} Q^k + C_2 \overline{Q^k}$$

b) $Q^k \rightarrow Q^{k+1}$ -re felírt táblázat alapján minden számítás nélkül:

$$T = \overline{C_1} Q^k + C_2 \overline{Q^k}$$

Abban a speciális esetben, amikor

$$\overline{C_1} = C_2, \quad T = \overline{C_1} = C_2.$$

A kapott eredményt ellenőrizzük $\overline{C_1} = C_2$ esetben. Mivel a T flip-flop logikai differencia egyenlete:

$$Q^{k+1} = \overline{T} Q^k + T \overline{Q^k}, \quad \text{ez } T = \overline{C_1} = C_2 \text{ helyettesítéssel a}$$

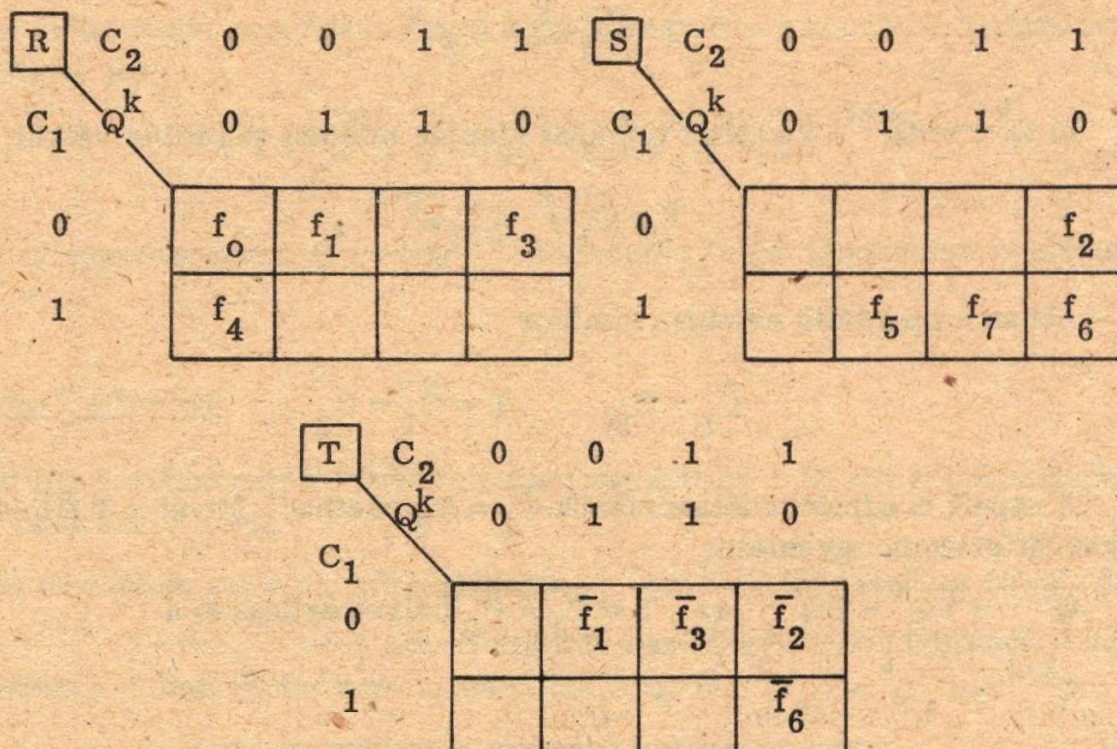
$$Q^{k+1} = C_1 Q^k + C_2 \overline{Q^k}$$

általános állapotegyenletből valóban leszármaztatható.

3.2.2.4. RST flip-flop

Mivel RST flip-flop esetén mind a redundáns mintermek, mind a lehetséges vezérlési lehetőségek száma nagyobb, mint az előzőekben tárgyalt tárolóknál, a bemenetek logikai egyenleteit csak a közvetlen módszerrel határozzuk meg.

index	C_1	C_2	Q^k	Q^{k+1}	R	S	T
0	0	0	0	0	f_0	0	0
1	0	0	1	0	f_1	0	$\bar{f}_1$
2	0	1	0	1	0	f_2	$\bar{f}_2$
3	0	1	1	0	f_3	0	$\bar{f}_3$
4	1	0	0	0	f_4	0	0
5	1	0	1	1	0	f_5	0
6	1	1	0	1	0	f_6	$\bar{f}_6$
7	1	1	1	1	0	f_7	0



3-18. ábra

RST tároló vezérlési táblázata, R, S, T bemenet vezérlési feltételének meghatározása Veitch diagram segítségével

A vezérlési táblázat (3-18. ábra) meghatározásánál figyelembe vettük az $RS = RT = ST = 0$ feltételeket. Lehetséges megoldások:

a) Amennyiben $f_0 = f_1 = f_6 = f_7 = 1$ és $f_2 = f_3 = f_4 = f_5 = 0$, akkor:

$$R = \overline{C_1} \overline{C_2}, \quad S = C_1 C_2, \quad T = \overline{C_1} C_2$$

b) $f_1 = f_2 = f_6 = 1$ és $f_3 = 0$, akkor:

$$R = \overline{C_1} \overline{C_2}, \quad S = C_2 \overline{Q^k}, \quad T = \overline{C_1} C_2 Q^k$$

c) $f_1 = f_3 = f_6 = 1$, $f_2 = 0$, akkor:

$$R = \overline{C_1} Q^k, \quad S = C_1 C_2, \quad T = \overline{C_1} C_2 Q^k$$

3.2.2.5. D flip-flop

Mivel a D flip-flop (késleltető elem) logikai egyenlete: $Q^{k+1} = D^k$, ezért

$$D = C_1 Q^k + C_2 Q^k$$

Amennyiben $\overline{C_1} C_2 = 0$, úgy $D = C_1 Q^k + C_2$, ha pedig $C_1 = C_2$, akkor $D = C_1$.

3.2.2.6. Összefoglalás

Kapott eredményeinket foglaljuk táblázatban össze:

Tipus	R ill. K	S ill. J	T	Feltétel
RS	$\bar{C}_1 Q^k$	$C_2 \bar{Q}^k$	-	$\bar{C}_1 C_2 \neq 0$
	$\bar{C}_1$	C_2		$\bar{C}_1 C_2 = 0$
JK	$\bar{C}_1$	C_2	-	-
	$\bar{C}_1 Q^k + \bar{Q}^k$	$C_2 \bar{Q}^k + \bar{Q}^k$	-	-
T			$\bar{C}_1 Q^k + C_2 \bar{Q}^k$	$\bar{C}_1 \neq C_2$
			$\bar{C}_1 = C_2$	$\bar{C}_1 = C_2$
RST	$\bar{C}_1 \bar{C}_2$	$C_1 C_2$	$\bar{C}_1 C_2$	-
	$\bar{C}_1 \bar{C}_2$	$C_2 \bar{Q}^k$	$\bar{C}_1 C_2 Q^k$	-
	$\bar{C}_1 Q^k$	$C_1 C_2$	$\bar{C}_1 C_2 Q^k$	
D	$D = C_1 Q^k + C_2 \bar{Q}^k$			-
	$D = C_1 Q^k + C_2$			$C_1 C_2 = 0$
	$D = C_1 = C_2$			$C_1 = C_2$
Általános alak	$Q^{k+1} = C_1 Q^k + C_2 \bar{Q}^k$			-
	$Q^{k+1} = C_1 Q^k + C_2$			$\bar{C}_1 C_2 = 0$
	$Q^{k+1} = C_1 = C_2$			$C_1 = C_2$

Tervezések során legtöbbször az általános megoldásból nyert bemeneti egyenleteket használjuk. A későbbiekben a táblázat eredményeire minden utalás nélkül hivatkozunk.

3.3 Gyakorló feladatok

1. Határozzuk meg az S, illetve R típusu tároló bemenetének vezérléséhez szükséges logikai egyenleteket a C_1 és C_2 vezérlési függvényekkel és Q^k -val kifejezve.

2. Határozzuk meg az inverz S, illetve inverz R típusu tároló bemenetének vezérléséhez szükséges logikai egyenleteket a C_1 és C_2 vezérlési függvényekkel és Q^k -val kifejezve. A tervezést végezzük el

a) Közvetlen módszer alkalmazásával.

b) $Q^k \rightarrow Q^{k+1}$ -re felírt táblázat segítségével.

3. Határozzuk meg az alábbi értelmezésű inverz működésű tárolók bemenetének vezérléséhez szükséges logikai egyenleteket a C_1 és C_2 vezérlési függvényekkel és Q^k -val kifejezve. A tervezést végezzük el

a) Közvetlen módszer alkalmazásával.

b) $Q^k \rightarrow Q^{k+1}$ -re felírt táblázat segítségével.

Bemenet		Tároló típusa				
R ill. K	S ill. J	RS	Inverz RS	Inverz JK	Inverz S	Inverz R
0	0	Q	?	$\bar{Q}$	1	0
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	1	?	Q	Q	Q	Q

4. T típusu tároló (Q) bemenetének vezérlésére szolgáló logikai egyenlet:

$$T_Q = BC + \bar{B}\bar{C} + \bar{A}Q$$

ahol: $ABC\bar{Q} = 0$. Határozzuk meg a bemenetek vezérlésére szolgáló egyenletet, ha a tárolót

a) D típusu

b) JK típusu

c) RS ill. RST

típusu tárolóval is meg akarjuk valósítani.

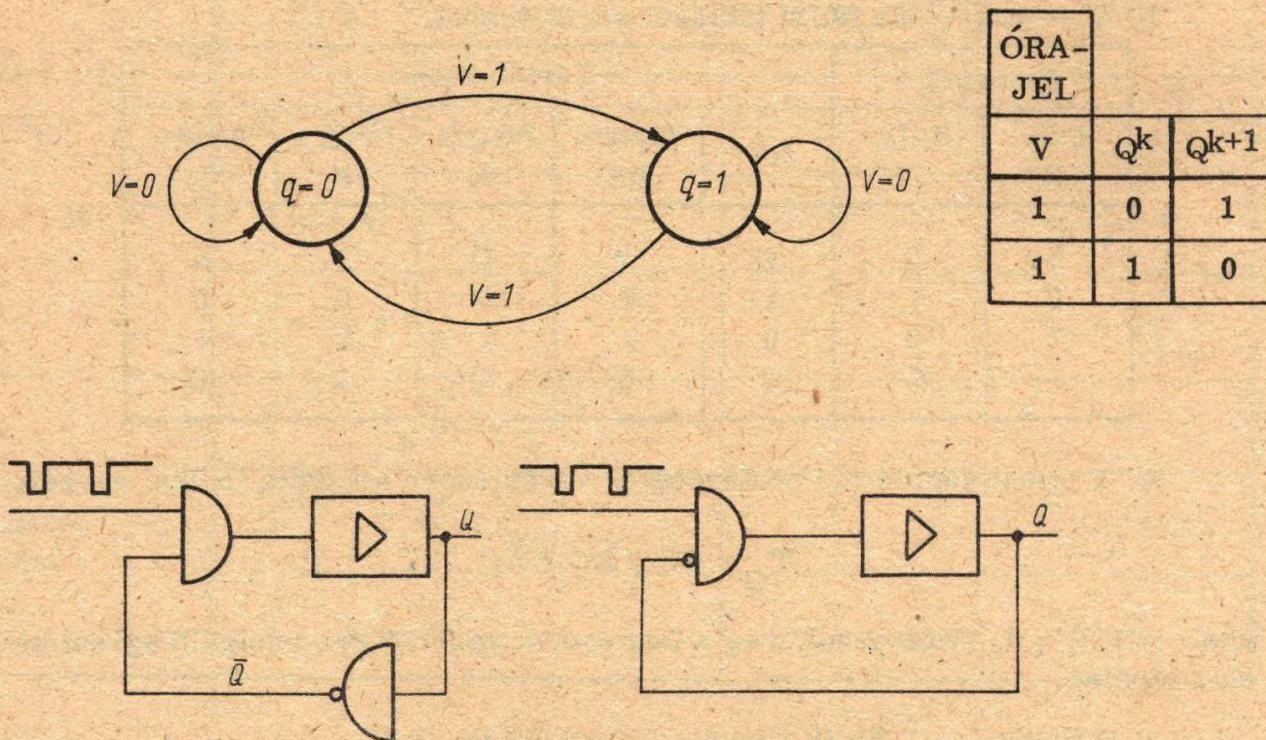
4. SEKVENCIÁLIS FUNKCIONÁLIS EGYSÉGEK

Ebben a fejezetben azokkal az egyszerűbb szekvenciális funkcionális egységekkel foglalkozunk, melyeknél a szekunder változók (belső állapotok) száma közvetlenül meghatározható.

4.1 Szekvenciális generátorok

Legegyszerűbb szekvenciális generátor az órajel ütemében 1, 0, 1, 0, 1, 0, ... stb. periodikusan ismétlődő bináris jelsorozatot állítja elő. Az áramkör állapot diagramjából vagy állapot táblázatából (4-1. ábra) leolvashatóan:

$$Q^{k+1} = \overline{Q^k}$$



4-1. ábra

1, 0, 1, 0, ... stb. jelsorozatot előállító bináris szekvenciális generátor

Az általános állapot egyenlettel összehasonlítva:

$$C_1 = 0, \quad C_2 = 1$$

A feladatot T típusu tárolóval megvalósítva, a bemenet vezérlési feltétele $(\overline{C_1} = C_2)$

$$T_A = \overline{C_1} = C_2 = 1$$

Kapott eredményünk a fizikai megfontolásokkal összhangban van, az órajelek hatására T típusu tároló mindig ellentétes állapotba billen.

RS típusu tároló alkalmazásakor $(\overline{C_1} C_2 \neq 0)$

$$G_0 = \overline{C_1} Q_k = Q_k, \quad G_1 = C_2 \overline{Q_k} = \overline{Q_k}$$

ami természetesen ismét T típusu tárolót eredményez.

JK típusu tároló esetén is ugyanezt az eredményt kapjuk:

$$K = \overline{C_1} = 1, \quad J = C_2 = 1$$

D típusu flip-flop (késleltető elem) felhasználásával:

$$D = C_1 Q^k + C_2 \overline{Q^k} = \overline{Q^k} = Q^{k+1}$$

Az áramkör (4-1. ábra) működése fizikailag könnyen követhető.

Tervezzük meg egy olyan bináris szekvenciális generátor logikai hatásvázlatát, amely az órajelek ütemében 00, 01, 11, 10, 00, 01, ... stb. periodikusan ismétlődő jelsorozatot állítja elő. Közvetlen szemlélet alapján belátható, hogy a feladat megoldásához két tároló elemet kell felhasználnunk. Ezek kimenő jelét A és B-vel jelölve az áramkör igazságtáblázata (4-2/a ábra) alapján:

$$\begin{aligned} A^{k+1} &= \overline{A^k} B^k + A^k \overline{B^k}, \quad \text{vagyis} \quad C_{1A} = C_{2A} = B^k \\ B^{k+1} &= \overline{A^k} \overline{B^k} + A^k B^k, \quad \text{vagyis} \quad C_{1B} = C_{2B} = \overline{A^k} \end{aligned}$$

T típusu tároló alkalmazásakor a bemeneti vezérlések:

$$\begin{aligned} T_A &= \overline{C_{1A}} A^k + C_{2A} \overline{A^k} = A^k \oplus B^k \\ T_B &= \overline{C_{1B}} B^k + C_{2B} \overline{B^k} = A^k \odot B^k \end{aligned}$$

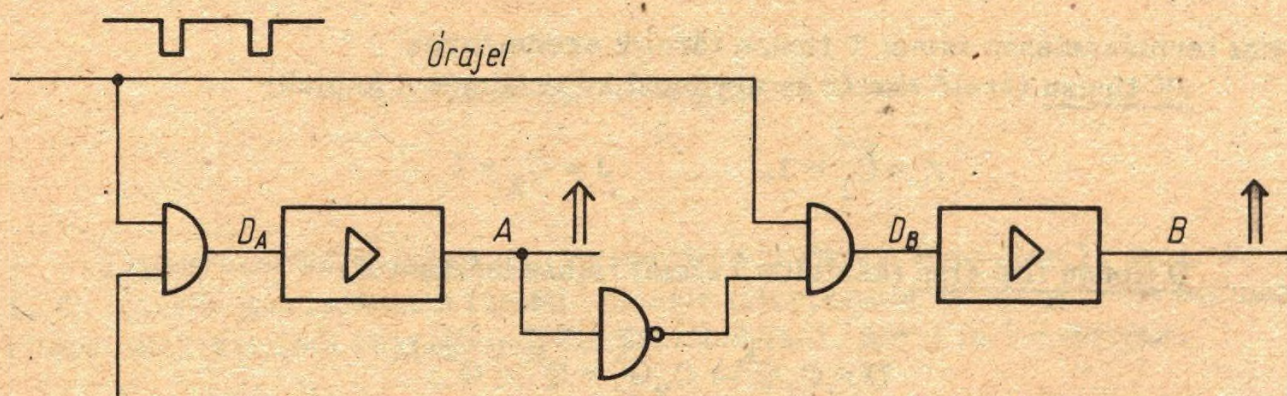
Az egyszerűsített logikai hatásvázlatokon az antivalencia és ekvivalencia kapukat elegendő csak jelképesen feltüntetni.

D típusú flip-flopok bemeneti vezérlése:

$$D_A = C_{1A} = B^k \quad \text{és} \quad D_B = C_{1B} = \overline{A^k}$$

A logikai hatásvázlatot (4-2/a ábra) tartalmazza.

V	A^k	B^k	A^{k+1}	B^{k+1}
1	0	0	0	1
1	0	1	1	1
1	1	1	1	0
1	1	0	0	0

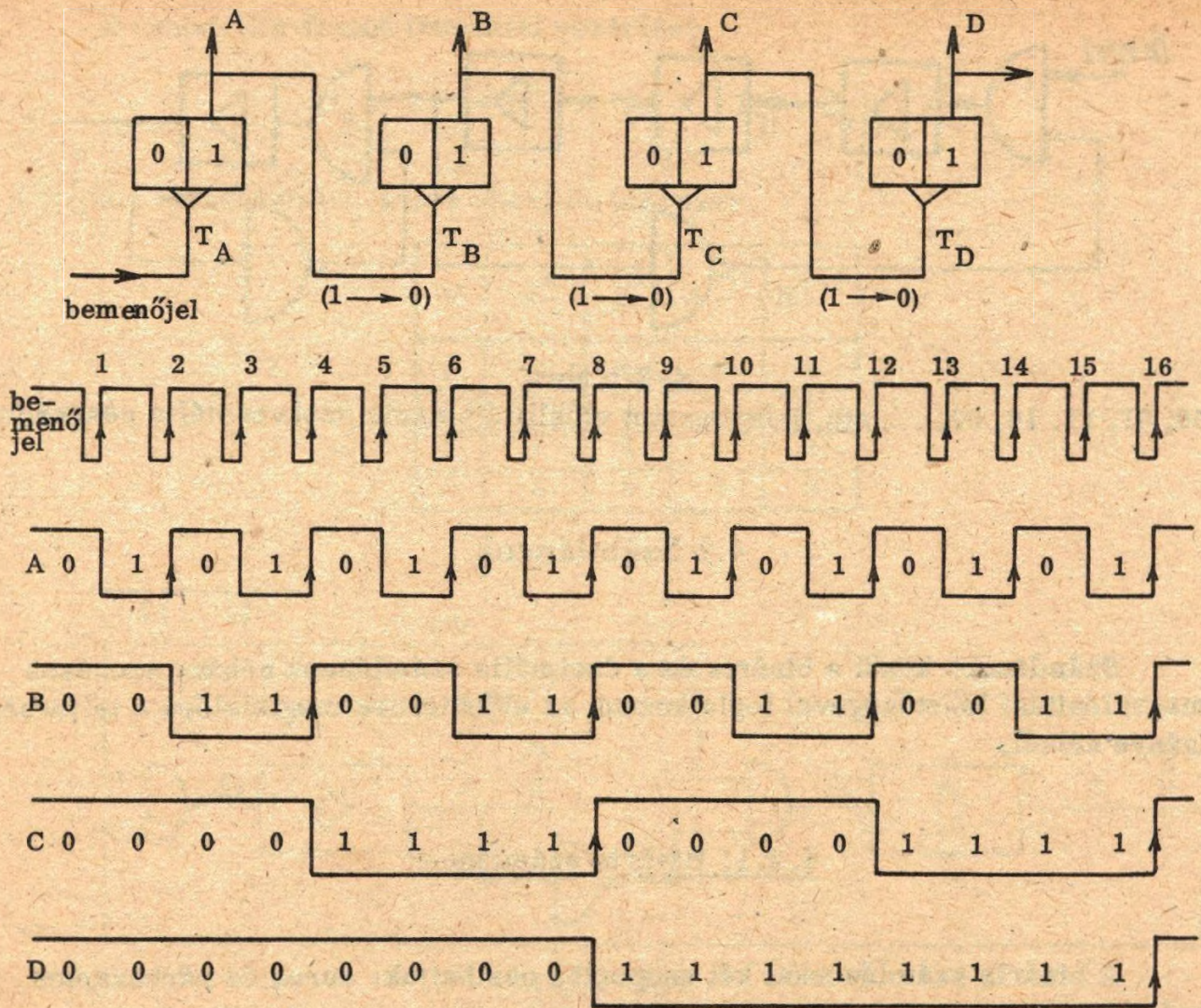


4-2/a ábra

00, 01, 11, 10, 00, ... stb. jelsorozat előállító
bináris szekvenciális generátor

4.1.1. Gyakorló feladatok

1. Határozzuk meg 4-2/b ábra bináris szekvenciális generátorának ki-menő jelét.
 2. Készítendő bináris szekvenciális generátor, mely V vezérlőjel hatásá-ra 100010001000... bináris jelsorozatot állítja elő.
 3. Tei vezendő RS típusú tárolók felhasználásával bináris szekvenciális generátor, amely az órajelek ütemében
000, 010, 011, 111, 110, 100, 000, 010, ... periodikusan ismétlődő bináris jelsorozatot állítja elő.
- Határozzuk meg a tárolók vezérlésének logikai egyenleteit. Rajzoljuk fel a műveletvégző regiszterek logikai hatásvázlatát.



4-3/a ábra

1 → 0 átmenetre billenő soros bináris számlánc és jelalakjai

T típusú tároló alkalmazása esetén ($\overline{C}_{1A} = C_{2A}$ és $\overline{C}_{1B} = C_{2B}$) bemeneti vezérlések:

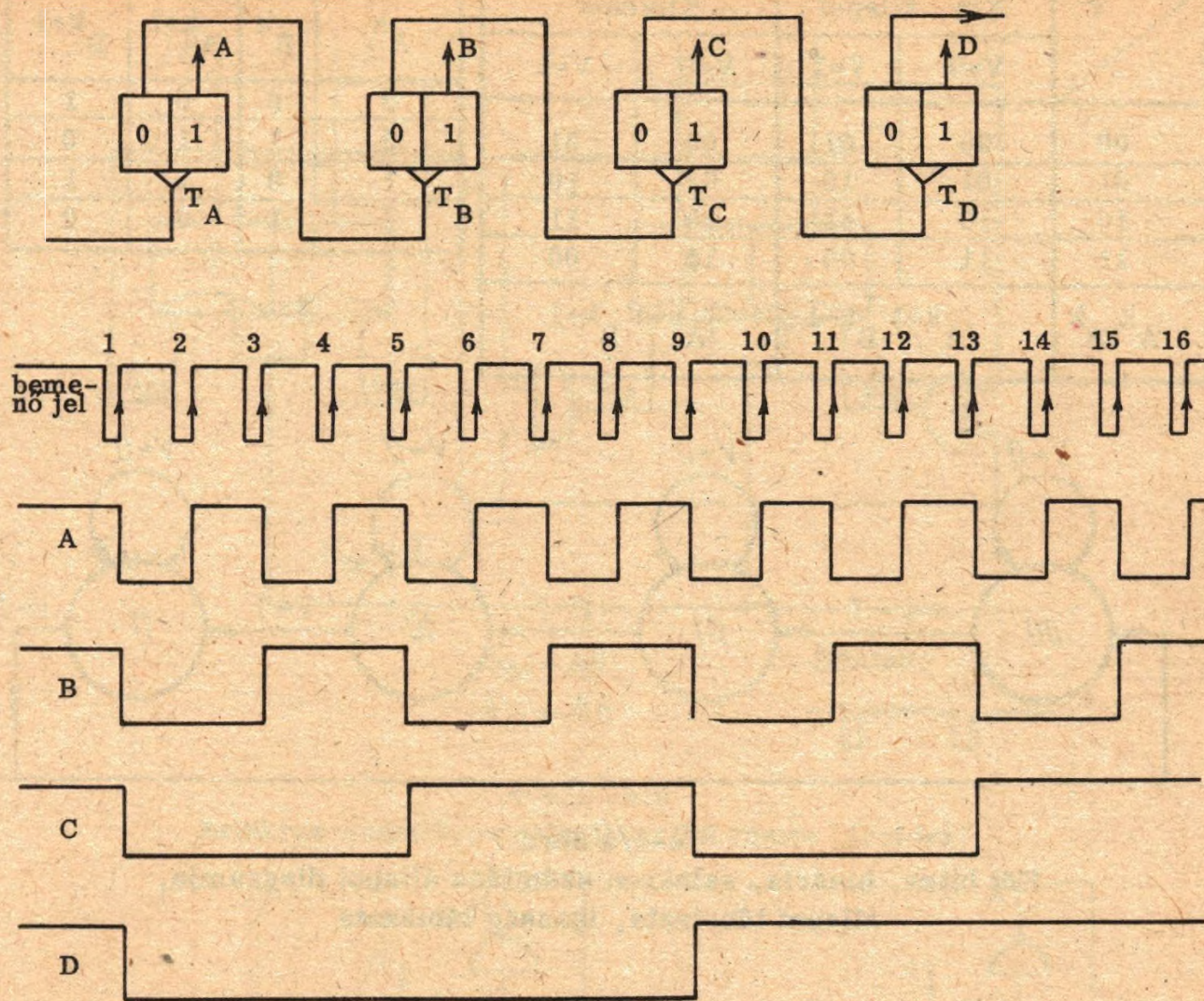
$$T_A = \overline{C}_{1A} = B^k, \quad T_B = \overline{C}_{1B} = 1$$

A T típusú tárolóval megvalósított két bites párhuzamos számlánc logikai hatásvázlatát a 4-4/b ábrán láthatjuk.

RS típusú tárolót felhasználva az A és B kimenetű tároló statikus kapuzó bemeneteinek vezérlése:

$$\begin{aligned} G_{oA} &= \overline{C}_{1A} A^k = A^k B^k, & G_{1A} &= C_{2A} \overline{A}^k = \overline{A}^k B^k \\ G_{oB} &= \overline{C}_{1B} B^k = B^k, & G_{1B} &= C_{2B} \overline{B}^k = \overline{B}^k \end{aligned}$$

A számlánc logikai hatásvázlatát a 4-4/c ábra tartalmazza.



4-3/b ábra

1→0 átmenetre billenő, inverz irányban számláló
soros bináris számláló és jelalakjai

Tervezzük meg a számlálót D flip-flop segítségével is (4-4/d ábra).
A késleltető elemek vezérlése:

és

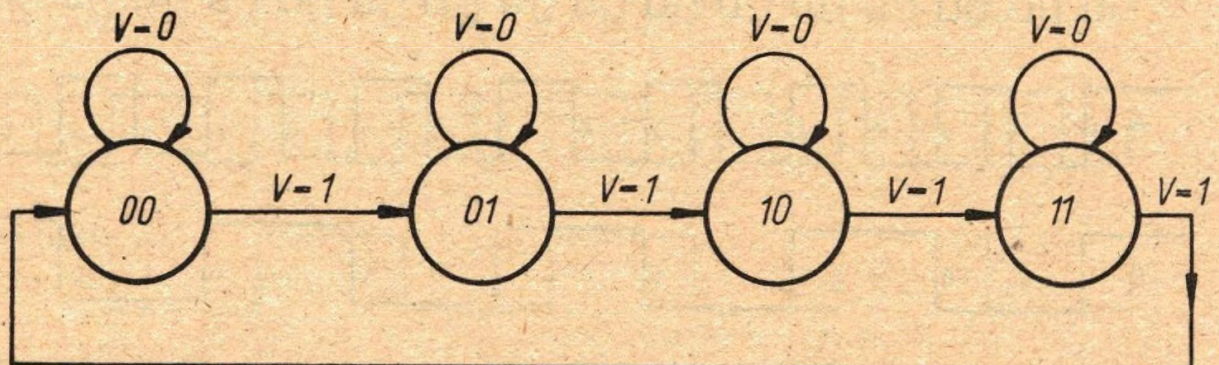
$$D_A = C_{1A} A^k + C_{2A} \overline{A^k} = A^k \oplus B^k$$

$$D_B = C_{1B} B^k + C_{2B} \overline{B^k} = \overline{B^k}$$

Gyakorlásképpen rajzoljuk fel a késleltető elemek be- és kimenő pontjain megjelenő jelalakokat az órajelek időpontjában! Elemezzük az áramkör működését fizikailag!

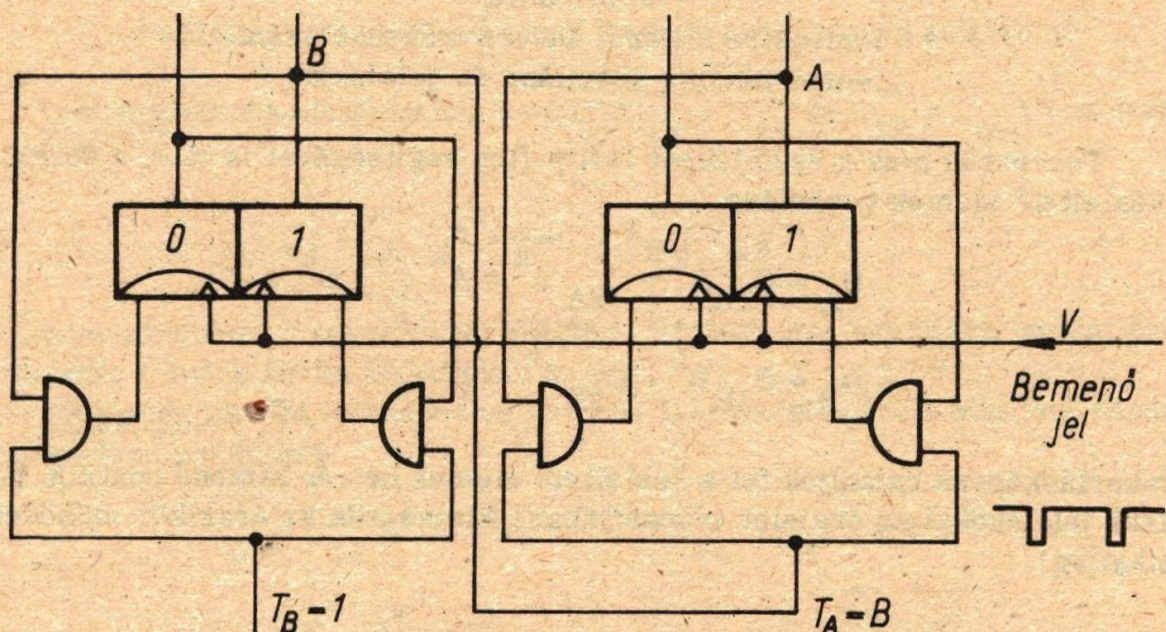
V \	Köv. állapot		Kimenet	
	V=0	V=1	V=0	V=1
00	00	01	00	01
01	01	10	01	10
10	10	11	10	11
11	11	00	11	00
$A^k B^k$	$A^{k+1} B^{k+1}$		$A^{k+1} B^{k+1}$	

A^k	B^k	A^{k+1}	B^{k+1}
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0



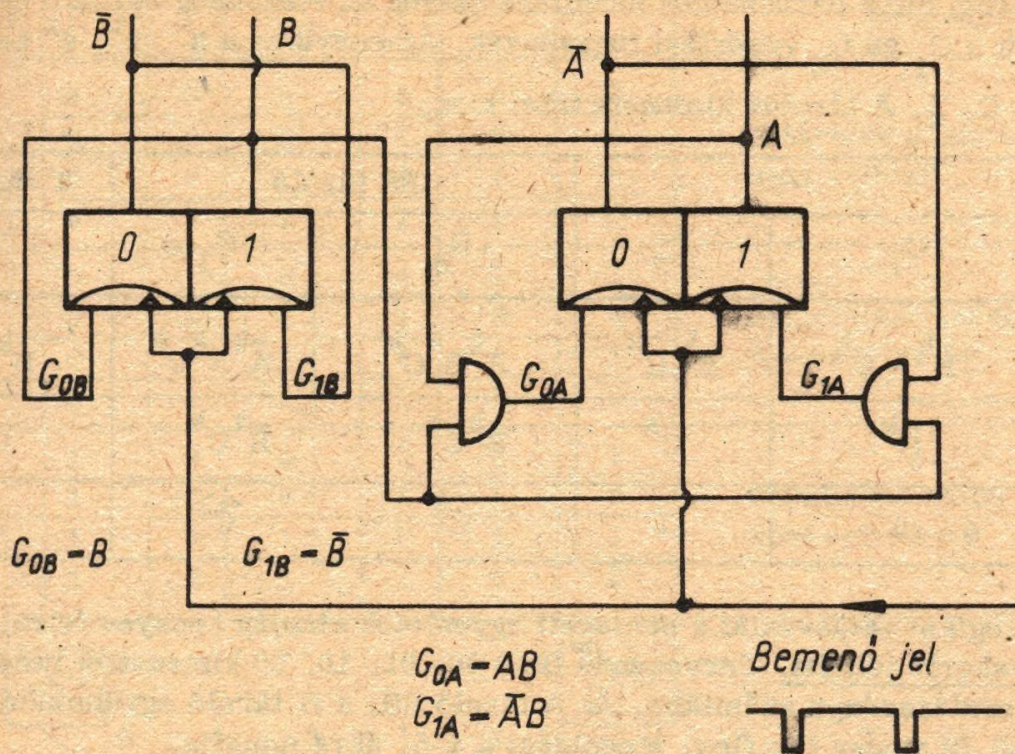
4-4/a ábra

Két bites, bináris, szinkron számláló állapot diagramja,
állapot táblázata, igazság táblázata



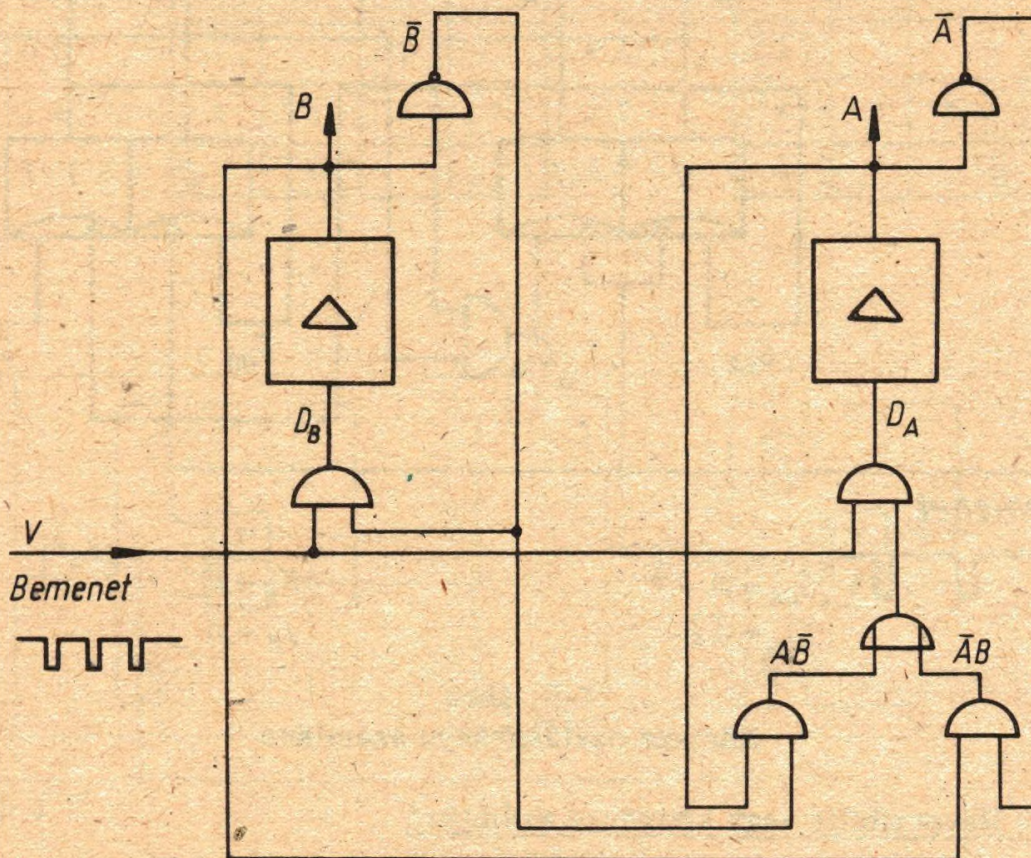
4-4/b ábra

Szinkron számláló realizálása T típusú tárolókkal



4-4/c ábra

Szinkron számlánc realizálása RS típusú tárolóval



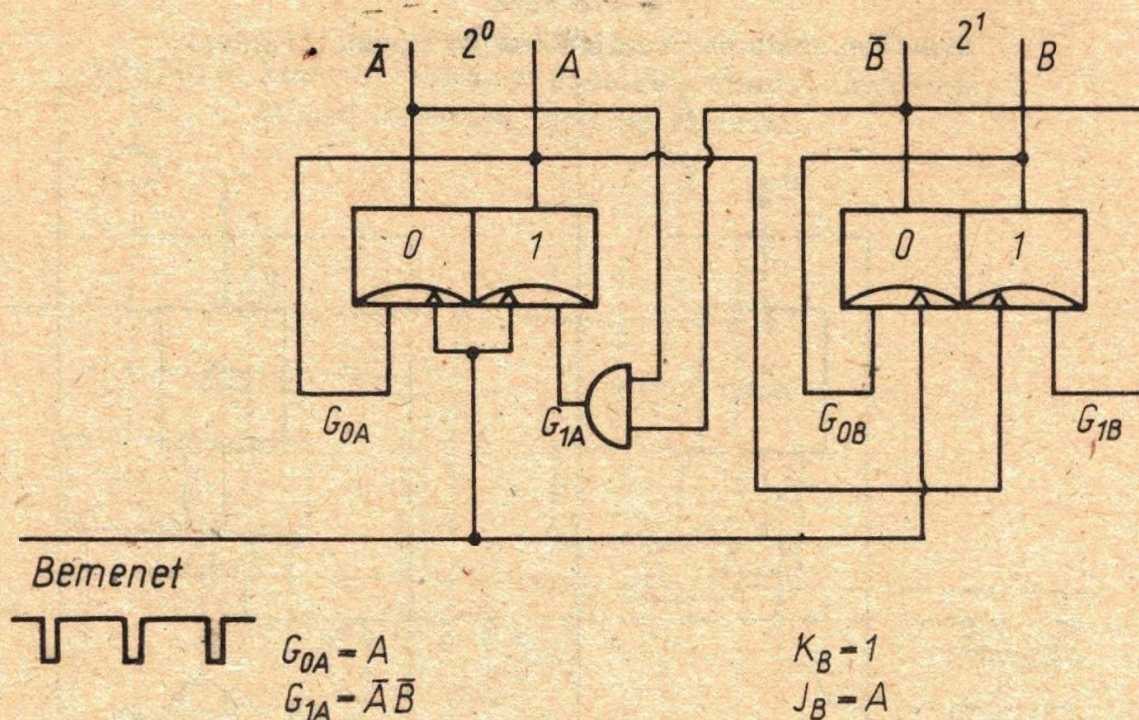
4-4/d ábra

Szinkron számlánc realizálása D típusú tárolóval

Három bites párhuzamos számlánc esetén az előzőkhöz hasonló módon nyerhetők a C_1 és C_2 vezérlési függvények. Amennyiben a 2^0 , 2^1 , 2^2 helyértékeknek a C, B, A tárolók kimenete felel meg:

			RS tároló		T tároló
	C_1	C_2	G_0	G_1	T
A	$\bar{B}^k + \bar{C}^k$	$B^k C^k$	$A^k B^k C^k$	$\bar{A}^k B^k C^k$	$B^k C^k$
B	$\bar{C}^k$	C^k	$B^k C^k$	$\bar{B}^k C^k$	C^k
C	0	1	C^k	$\bar{C}^k$	1

A logikai hatásvázlat a bemeneti egyenletek alapján könnyen felrajzolható.
Gyakorló feladat. Tervezendő (BA) 00, 01, 10, 00 kimenettel rendelkező hármas osztásarányú számlánc. Az A tároló RS, a B tároló együtemű dinamikus vezérlésű JK típusú flip flop. Megoldást a 4-5. ábra mutatja.

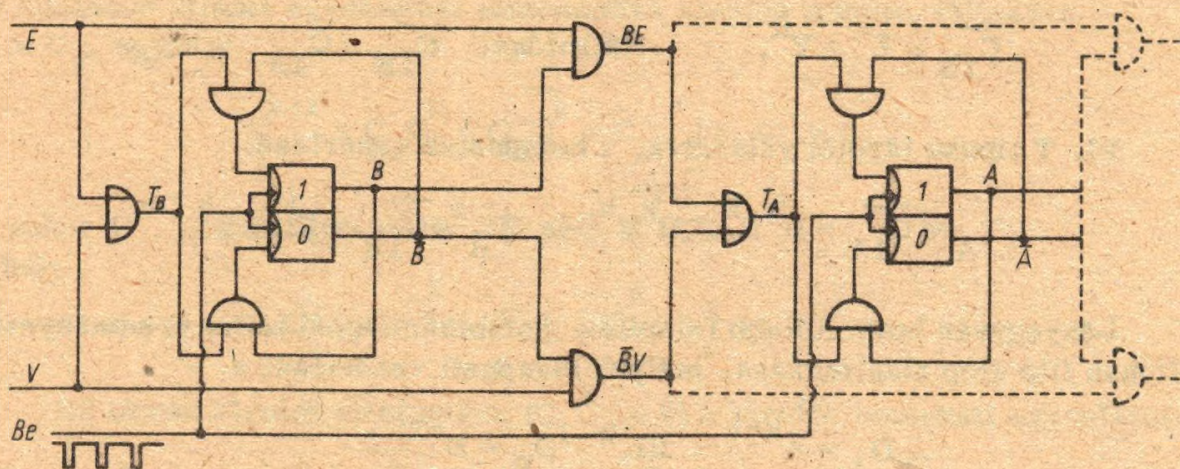
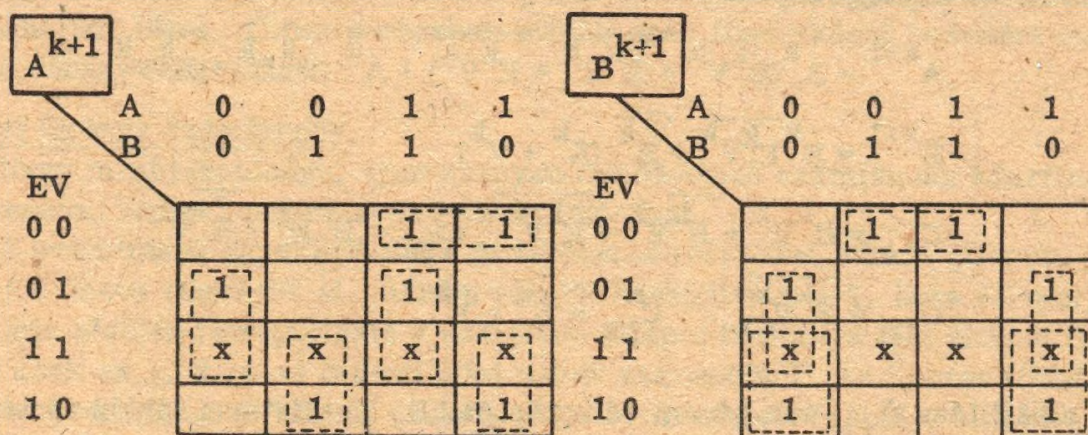


4-5. ábra
Hármas osztásarányú számlánc

c) Reverzibilis vagy kétirányú számlánc

A számítások egyszerűsítése végett két tárolóval megvalósított reverzibilis számlánc tervezését mutatjuk be.

index	E	V	A^k	B^k	A^{k+1}	B^{k+1}	Megjegyzés
0	0	0	0	0	0	0	A tárolók állapota nem változik
1	0	0	0	1	0	1	
2	0	0	1	0	1	0	
3	0	0	1	1	1	1	
4	0	1	0	0	1	1	Számlálás vissza $0 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$
5	0	1	0	1	0	0	
6	0	1	1	0	0	1	
7	0	1	1	1	1	0	
8	1	0	0	0	0	1	Számlálás előre $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$
9	1	0	0	1	1	0	
10	1	0	1	0	1	1	
11	1	0	1	1	0	0	



4-6. ábra
Reverzibilis számláló igazságtáblázata, minterm táblái, realizálása

Az "E" (előre) vezetéken érkező statikus vezérlőjel hatására a számlánc (AB) 00, 01, 10, 11, 00, ... stb. sorrendben, a "V" (vissza) bemenetre érkező statikus vezérlőjel hatására 00, 11, 10, 01, 00, ... stb. sorrendben számlál. A tervezéshez a két bemeneti változón (E, V) kívül két szekunder változóra van szükségünk (A, B). Mivel az $E = V = 1$ eset vezérlés során nem fordulhat elő, vagyis $EV = 0$, a négy változós igazságtáblán a 12, 13, 14, 15 indexű mintermek együtthatója szabadon felvehető (4-6. ábra).

A C_1, C_2 vezérlési függvények meghatározásához az

$$A^{k+1} = m_2 + m_3 + m_4 + m_7 + m_9 + m_{10}$$

és

$$B^{k+1} = m_1 + m_3 + m_4 + m_6 + m_8 + m_{10}$$

négy változós mintermeket Veitch diagramon ábrázoljuk.

A minimalizált függvények:

$$A^{k+1} = A^k (B^k V^k + \overline{B^k E^k} + \overline{E^k V^k}) + \overline{A^k} (B^k E^k + \overline{B^k V^k})$$

$$B^{k+1} = B^k \overline{E^k V^k} + \overline{B^k} (E^k + V^k)$$

$$C_{1A} = B^k V^k + \overline{B^k E^k} + \overline{E^k V^k}, \quad C_{2A} = B^k E^k + \overline{B^k V^k}$$

$$C_{1B} = \overline{E^k V^k}, \quad C_{2B} = E^k + V^k$$

$\overline{C_{1A}}$ előállítására C_{1A} -nak három változós (B, E, V) minterm táblán történő ábrázolásával történhet:

$$\overline{C_{1A}} = B^k E^k + \overline{B^k V^k}, \quad \text{láthatóan: } \overline{C_{1A}} = C_{2A}$$

$$\overline{C_{1B}} = E^k + V^k, \quad \text{láthatóan: } \overline{C_{1B}} = C_{2B}$$

Pl. T típusú tárolót választva, a bemenetek vezérlése:

$$T_A = \overline{C_{1A}} = B^k E^k + \overline{B^k V^k} \quad \text{és} \quad T_B = \overline{C_{1B}} = E^k + V^k$$

Lényegesen bonyolultabb felépítésű kombinációs hálózatot eredményezne D típusú flip-flop alkalmazása, melyek bemeneti vezérlését a

$$D_A = A^{k+1} \quad \text{ill.} \quad D_B = B^{k+1}$$

egyenletek adnák.

A számláló a közös dinamikus bemeneten (egyetlen bemenet) érkező impulzusokat számlálja, két lehetséges üzemmódban. Az üzemmód beállítása külső vezérléssel történik (E ill. V vezérlőjel alkalmazásával). A számlánc a két üzemmódban érkező impulzusok számának különbségét tárolja.

Készíthető két független számláló (impulzus) bemenettel rendelkező differencia számláló is. Az egyik bemeneten érkező impulzusok hatására előre számlál, a másik bemeneten érkező impulzusok hatására visszafelé számlál. A számlánc ebben az esetben is a két bemeneten érkező impulzusok számának különbségét tárolja.

4.2.2. Decimális számláncok

Decimális számláncot részben a megvalósítás módja, részben a választott kódrendszer (1-2-4-8-as bináris kód, 1-2-4-4-es kód, 5-ből 2-es kód, 1-2-4-2-es Aiken kód, Stibitz kód stb.) tulajdonságai alapján igen sokféleképpen hozhatunk létre. Ebben a részben inkább csak illusztráció kedvéért mutatunk be néhány megoldást.

a) Gyűrűs számláncok.

Nagy megbízhatóságu, igen könnyen dekódolható számláncok készíthetők visszacsatolt léptető regiszterek felhasználásával.

A számítások egyszerűsítése érdekében a tervezést három állapotú gyűrűs számláncra végezzük el. Három tárolót használva a 4-7. ábra állapot diagramja alapján felrajzolható a számlánc állapot táblázata. Az állapot táblázatban csak az 1, 2, 4-es indexű mintermek szerepelnek, az összes többi minterm redundáns. A három szekunder változó minterm táblája alapján:

$$A^{k+1} = C^k, \quad B^{k+1} = A^k, \quad C^{k+1} = B^k$$

Az általános állapot egyenletnél elmondottak alapján minden változóra $C_1 = C_2$, ugyanis

$$Q^{k+1} = C_1 Q^k + C_2 \overline{Q^k} = C_1, \quad \text{ha } C_1 = C_2$$

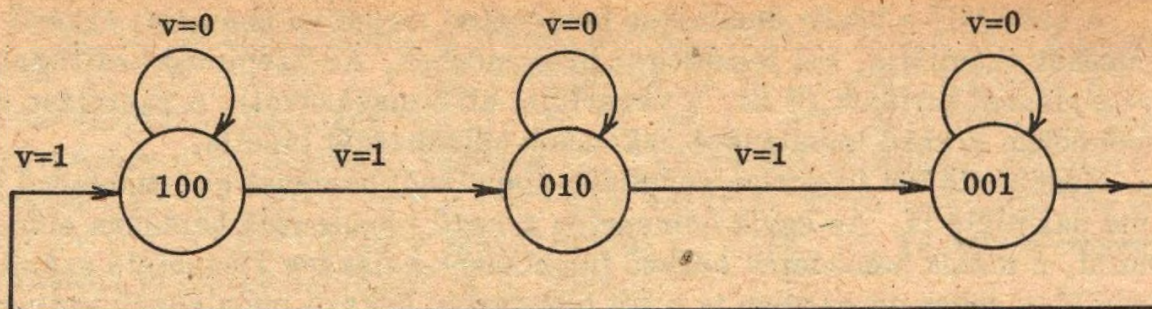
és ekkor Q^{k+1} nem függ sem Q^k -től sem $\overline{Q^k}$ -től.

Tehát:

$$C_{1A} = C^k, \quad C_{1B} = A^k, \quad C_{1C} = B^k$$

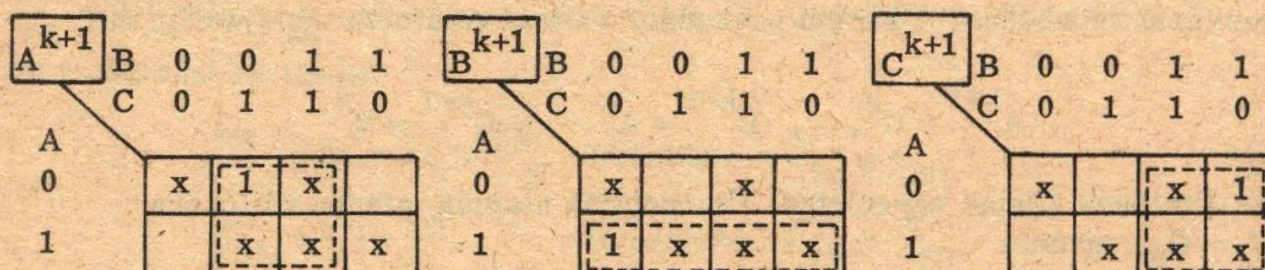
RS típusú tárolót alkalmazva ($\overline{C_1} C_2 = 0$) a tárolók bemeneti egyenletei:

$$\begin{array}{lll} G_{0A} = \overline{C^k} & G_{0B} = \overline{A^k} & G_{0C} = \overline{B^k} \\ G_{1A} = C^k & G_{1B} = A^k & G_{1C} = B^k \end{array}$$



$A^k B^k C^k \backslash V$	Köv. állapot		Kimenet	
	V = 0	V = 1	V = 0	V = 1
1 0 0	1 0 0	0 1 0	1 0 0	0 1 0
0 1 0	0 1 0	0 0 1	0 1 0	0 0 1
0 0 1	0 0 1	1 0 0	0 0 1	1 0 0
	$A^{k+1} B^{k+1} C^{k+1}$		$A^{k+1} B^{k+1} C^{k+1}$	

index	A^k	B^k	C^k	A^{k+1}	B^{k+1}	C^{k+1}
4	1	0	0	0	1	0
2	0	1	0	0	0	1
1	0	0	1	1	0	0



4-7. ábra
Gyűrűs számlánc állapot diagramja, állapot táblázata,
igazságtáblázata és minterm táblái

A háromfokozatu gyűrűs regiszter tehát 3 bites léptető regiszter visszacsatolásával nyerhető.

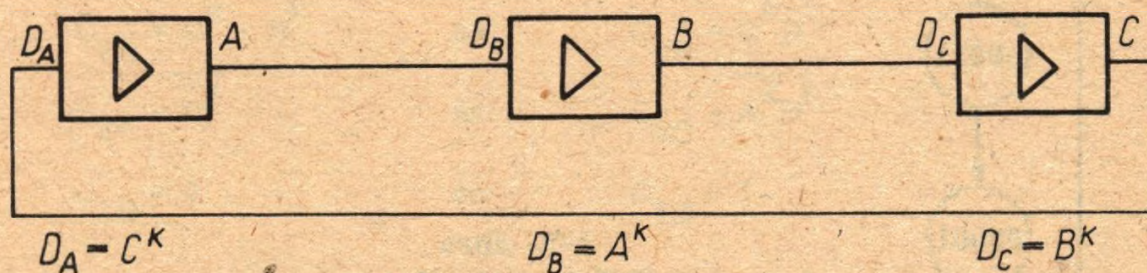
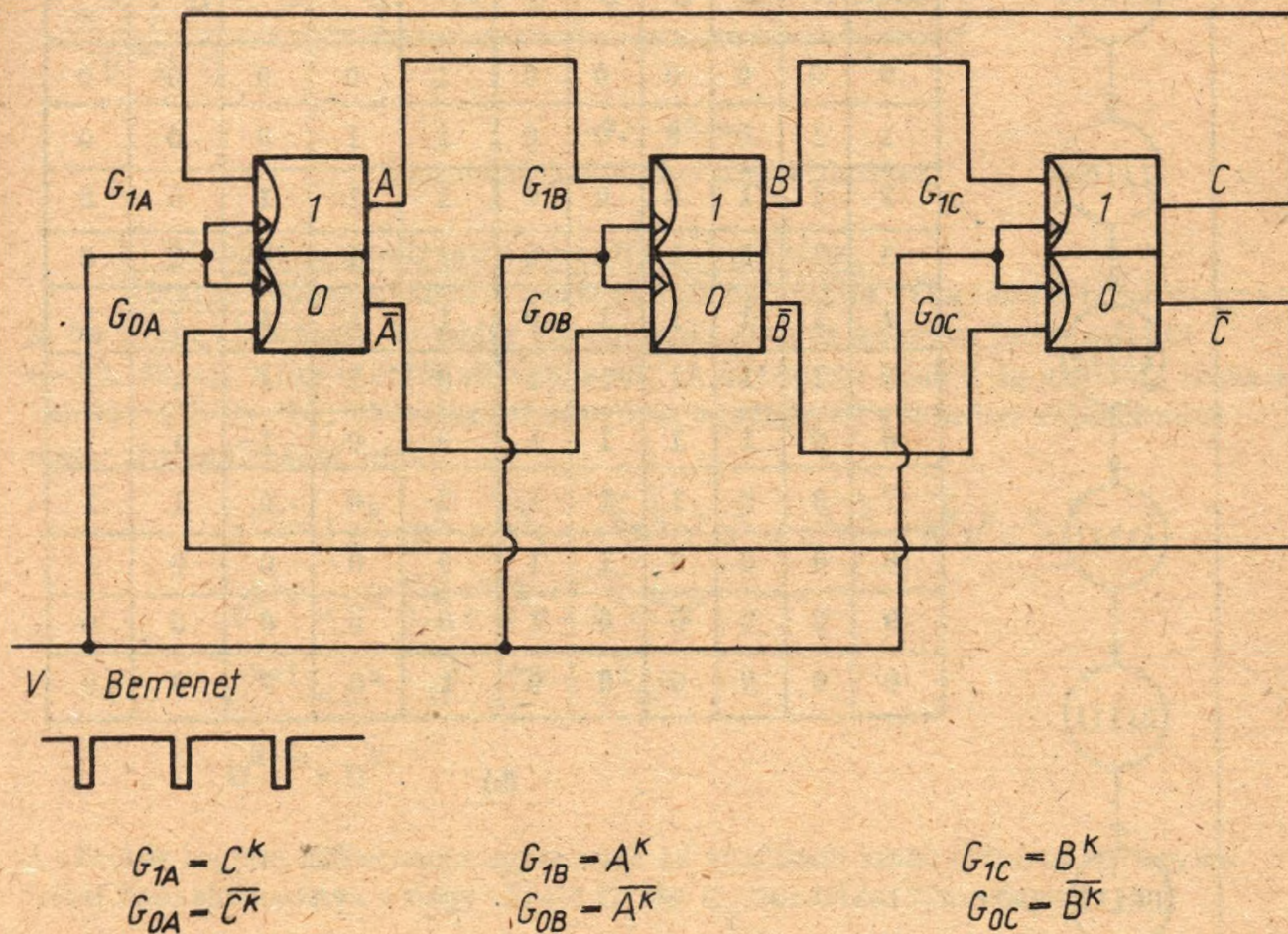
D típusu flip-flop alkalmazásával a késleltető elemek szerepe jól látható:

$$D_A = C_{1A} = C^k, \quad D_B = A^k, \quad D_C = B^k$$

(4-8. ábra).

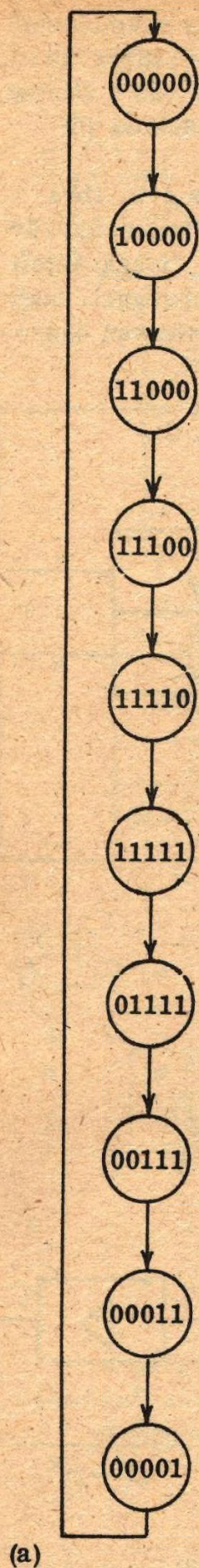
A fenti elven 10 tároló alkalmazásával készíthetnénk gyűrűs számláncból decimális számláncot. A számlánc a 10-ből 1 vagy más néven a 0123456789 súlyozási tényezőjű additív kódban számol. (Mindig csak 1 helyen van 1 állapot, viszont a kódszó 10 helyértékből áll.) Pl. lyukkártyás berendezésekben jól használható ez a nagy redundanciájú kód.

A tárolók fele megtakarítható, ha a számlánc 10 állapotát a 4-9. ábra állapot diagramja alapján kódoljuk. Az állapot táblázat alapján a tervezés már könnyen elvégezhető. A megoldás az ábrán láthatóan most is egy visszacsatolt léptető regiszterből (vagy gyűrűs regiszterből) áll, azzal a különbséggel, hogy a gyűrűs regiszter eleje és vége a 4-8. ábrához képest ellentétesen van összekötve.



4-8. ábra

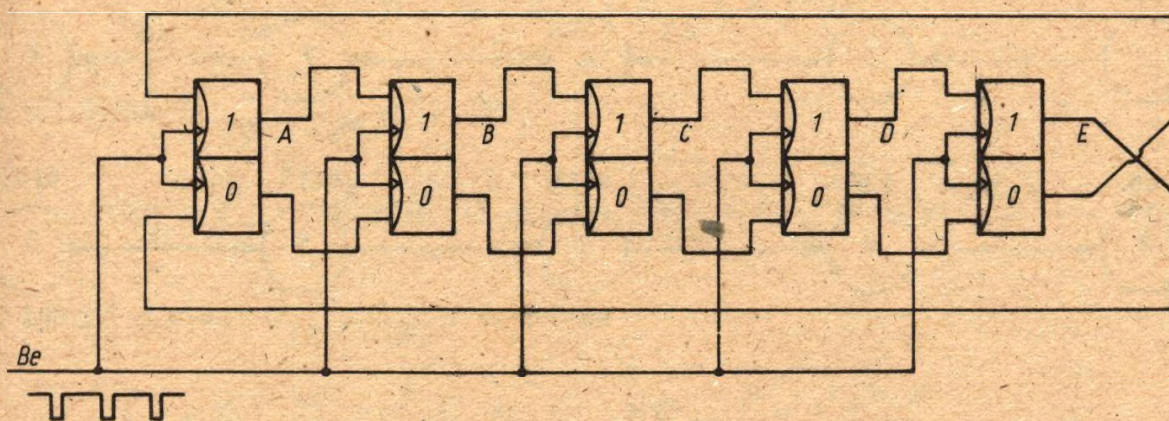
Gyűrűs számlánc realizálása RS és D típusú tárolóval



(b)

imp.	A^k	B^k	C^k	D^k	E^k	A^{k+1}	B^{k+1}	C^{k+1}	D^{k+1}	E^{k+1}
0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	1	1	0	0	0
2	1	1	0	0	0	1	1	1	0	0
3	1	1	1	0	0	1	1	1	1	0
4	1	1	1	1	0	1	1	1	1	1
5	1	1	1	1	1	0	1	1	1	1
6	0	1	1	1	1	0	0	1	1	1
7	0	0	1	1	1	0	0	0	1	1
8	0	0	0	1	1	0	0	0	0	1
9	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0

4-9. ábra
Gyűrűs számláló
a) állapot diagramja
b) igazságtáblázata



4-9/c ábra
Decimális számlánc megvalósítása

b) Párhuzamos beírási decimális számláncok.

Bináris kódban számláló párhuzamos decimális számlánc állapot diagramja és állapot táblázata az eddig elmondottak alapján könnyen felrajzolható (4-10. ábra). Négy tárolót alkalmazunk. A számlánc csak 9-ig számol, tehát a 10, 11, 12, 13, 14, 15 indexű mintermek együttthatói szabadon választhatók meg. Minimalizálás után:

$$A^{k+1} = A^k (B^k + \overline{D}^k) + \overline{A}^k B^k C^k D^k = A^k \overline{D}^k + \overline{A}^k B^k C^k D^k$$

$$B^{k+1} = B^k (\overline{C}^k + \overline{D}^k) + \overline{B}^k C^k D^k$$

$$C^{k+1} = C^k \cdot \overline{D}^k + \overline{C}^k \cdot \overline{A}^k D^k$$

$$D^{k+1} = \overline{D}^k$$

A tárolók logikai differencia egyenletét az általános alakú differencia egyenlettel összehasonlítva a négy tároló C_1 és C_2 vezérlési függvénye:

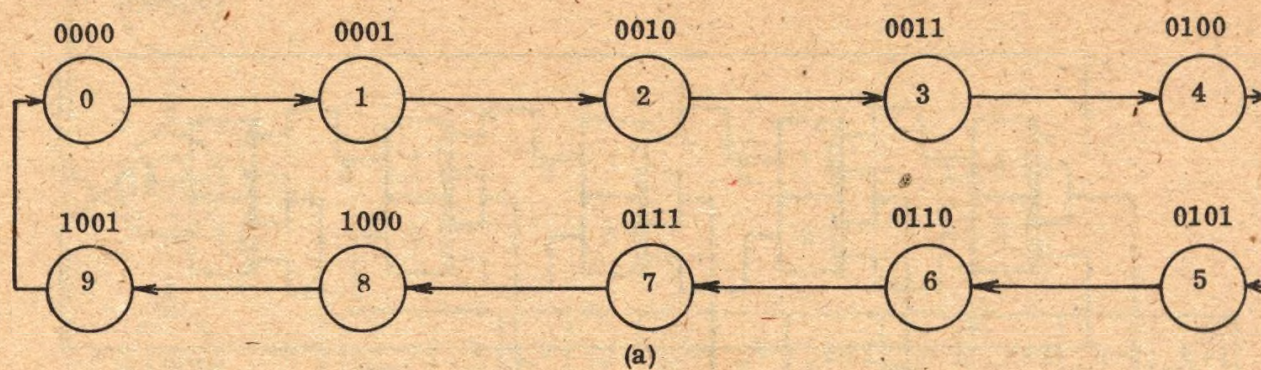
$$C_{1A} = B^k + \overline{D}^k \quad \text{és} \quad C_{2A} = B^k C^k D^k$$

$$C_{1B} = \overline{C}^k + \overline{D}^k \quad \text{és} \quad C_{2B} = C^k D^k$$

$$C_{1C} = \overline{D}^k \quad \text{és} \quad C_{2C} = \overline{A}^k D^k$$

$$C_{1D} = 0 \quad \text{és} \quad C_{2D} = 1$$

Valósítsuk meg a binárisan kódolt decimális számrendszerű számláncot RS tárolókkal. A logikai hatásvázlat felrajzolásához a G_0 és G_1 bemenet vezérlését kell meghatározni.



index	A^k	B^k	C^k	D^k	A^{k+1}	B^{k+1}	C^{k+1}	D^{k+1}
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
10	1	0	1	0	x	x	x	x
11	1	0	1	1	x	x	x	x
12	1	1	0	0	x	x	x	x
13	1	1	0	1	x	x	x	x
14	1	1	1	0	x	x	x	x
15	1	1	1	1	x	x	x	x

(b)

4-10. ábra

Bináris kódban számláló decimális számlánc
(a) állapot diagramja, (b) igazságtáblázata

- Mivel: $\overline{C_{1A}} \cdot C_{2A} = 0$, ezért

$$G_{0A} = \overline{C_{1A}} = \overline{B^k} \cdot D^k \text{ és } G_{1A} = C_{2A} = B^k C^k D^k$$

- B tárolónál: $\overline{C_{1B}} \cdot C_{2B} \neq 0$, így

$$G_{0B} = \overline{C_{1B}} \cdot B^k = \overline{B^k} \cdot C^k \cdot D^k \text{ és } G_{1B} = C_{2B} \cdot \overline{B^k} = \overline{B^k} C^k D^k$$

- C tároló esetén: $\overline{C_{1C}} \cdot C_{2C} \neq 0$, ezért

$$G_{0C} = \overline{C_{1C}} \cdot C^k = C^k D^k \text{ és } G_{1C} = C_{2C} \cdot \overline{C^k} = \overline{A^k} C^k D^k$$

- Végül a D tárolónál: $\overline{C_{1D}} \cdot C_{2D} \neq 0$, tehát

$$G_{0D} = \overline{C_{1D}} \cdot D^k = D^k \text{ és } G_{1D} = C_{2D} \cdot \overline{D^k} = \overline{D^k}$$

A számlánc realizálására T típusú tárolókat használva a tárolók bemeneti vezérlésének meghatározásához a $\overline{C_1} = C_2$ egyenlőséget kell megvizsgálnunk.

- Az A tárolónál: $\overline{C_{1A}} \neq C_{2A}$ ezért

$$T_A = \overline{C_{1A}} A^k + C_{2A} \overline{A^k} = A^k \cdot D^k + \overline{A^k} B^k \cdot C^k D^k = A^k D^k + B^k C^k D^k$$

- B tároló esetén: $\overline{C_{1B}} = C_{2B}$, ennek megfelelően

$$T_B = \overline{C_{1B}} = C^k \cdot D^k$$

- C tároló vezérlési függvényeiből

$$\overline{C_{1C}} \neq C_{2C},$$

minimalizálás után:

$$T_C = \overline{A^k} D^k$$

- Végül a D tároló bemeneti vezérlésének meghatározásánál:

$$\overline{C_{1D}} = C_{2D},$$

tehát

$$T_D = \overline{C_{1D}} = 1$$

Bináris kódban számláló decimális számlánc megvalósítása T típusú tárolókkal

Ebben a rövid összefoglalásban célunk nem a minimális elemszám (pl. dióda darabszám) meghatározása volt, hanem a tervezés elvi menetének bemutatása. Láthatóan a határozatlan (redundáns) kombinációk (10, 11, 12, 13, 14, 15-ös indexű mintermek decimális számlánc esetén) a realizálást szekvenciális hálózatoknál is egyszerűsítik, ezeket a bemeneti egyenletek meghatározásánál figyelembe kell venni.

01100 → 00110 → 00011 → 10001 → 11000 → 01010 →
→ 00101 → 10010 → 01001 → 10100 → 011000 → stb.

- 82 -

$$D_A = Q_E^k; \quad D_B = Q_A^k; \quad D_C = \overline{Q_A^k} Q_B^k + Q_A^k Q_C^k;$$

$$D_D = \overline{Q_A^k} Q_C^k + Q_A^k Q_B^k; \quad D_E = Q_D^k$$

Tekintettel arra, hogy D típusú tároló esetén:

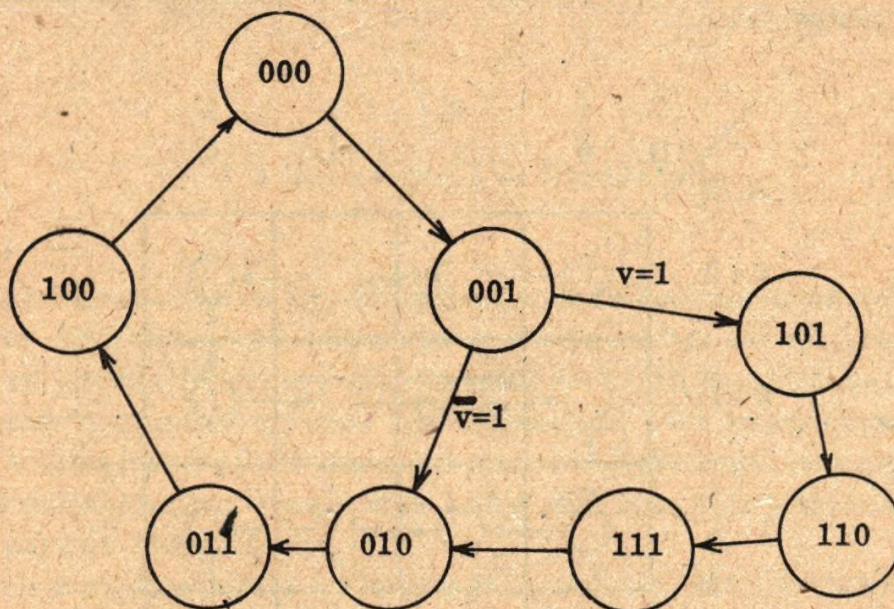
$$D = C_1 Q^k + C_2 \overline{Q^k}$$

a C_1 és C_2 vezérlési függvény minden tárolónál közvetlenül meghatározható, amiből más típusú tároló bemeneti vezérlése is felírható.

4.2.3. Gyakorló feladatok

1. Számítógépek műveletszámláló egysége számára tervezendő 4-12/a ábrán feltüntetett állapotdiagrammal rendelkező sorozatszámoló. A sorozatszámoló 001 állapot elérése után a V vezérlőjel 1-es értéke esetén
 001 → 101 → 110 → 111 → 010 → 011 → 100 → 000 → 001, $V = 0$ esetén:
 001 → 010 → 011 → 100 → 000 → 001 sorrendben számol.

A sorrendszámlálót T, ill. RS típusú tároló segítségével realizáljuk. Írjuk fel a tárolók vezérlésének bemeneti egyenleteit.



4-12/a ábra

2. Tervezendő decimális számlánc RS, T, JK típusú tárolók felhasználásával

a) 2421 (Aiken) kódban

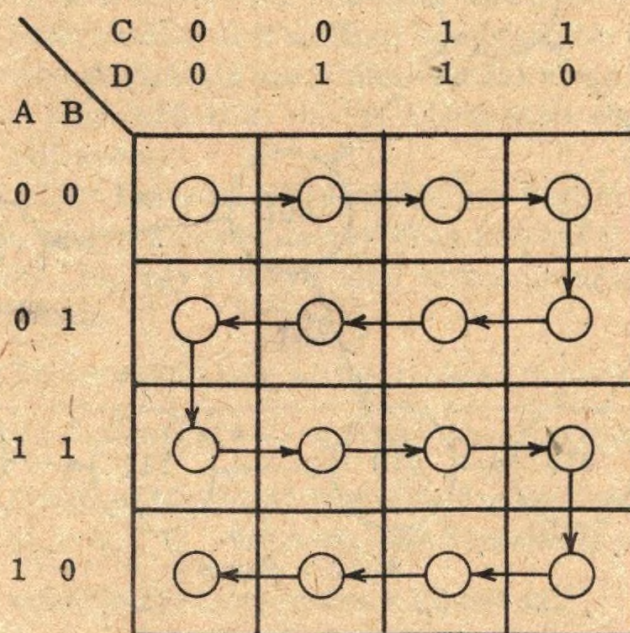
b) excess three (három többletes) vagy más néven Stibitz kódban.

Meghatározandók a tárolók bemeneti egyenletei.

3. Tervezendő Gray kódban számláló decimális számlánc T típusú tárolók felhasználásával. A tárolók állapot változását tartalmazó állapot táblázat:

Álla- pot	k-adik ütem				(k+1)-edik ütem			
	A^k	B^k	C^k	D^k	A^{k+1}	B^{k+1}	C^{k+1}	D^{k+1}
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	1
2	0	0	1	1	0	0	1	0
3	0	0	1	0	0	1	1	0
4	0	1	1	0	0	1	1	1
5	0	1	1	1	0	1	0	1
6	0	1	0	1	0	1	0	0
7	0	1	0	0	1	1	0	0
8	1	1	0	0	1	1	0	1
9	1	1	0	1	0	0	0	0

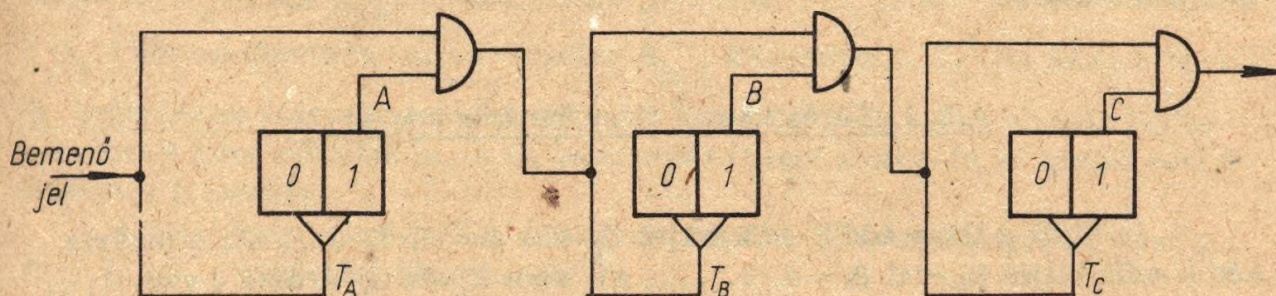
4. Tervezendő Gray kódban számláló 16 állapotu (modulo -2^4) számlánc T, ill. JK típusú tárolók felhasználásával. A számlánc állapot változásait a 4-12/b ábra tünteti fel.



4-12/b ábra

Készítsük el a Gray kódban számláló számlánc igazságtáblázatát, határozzuk meg a tárolók vezérlésének bemeneti egyenleteit.

5. Vizsgáljuk meg 4-12/c ábrán feltüntetett kapuzott számlánc működését. A tárolók 1→0 átmenetre billenő T típusu flip-floppok. Rajzoljuk fel a tárolók 1-es kimenetéhez tartozó jelalakokat.



4-12/c ábra

6. Tervezendő decimális számlánc, amely a 9-ik impulzus beérkezéséig

- binárisan kódolt decimális számrendszerben
- Gray kódban

számlál, külső vezérléssel előre beállíthatóan. A 10. beérkező impulzus hatására alaphelyzetbe tér vissza, melyben minden tároló 0-ás állapotban van. A számlánc V_T vezérlőjellel bármely állapotban nullázható.

A felhasznált tárolók típusa: T.

Meghatározandó a számlánc állapot diagramja, állapot táblázata a törlés lehetőségének feltüntetésével. Felrajzolandó az igazságtáblázat. Irjuk fel a tárolók bemeneti egyenleteit.

4.3 Szekvenciális átalakítók

Az egy bemenettel és egy kimenettel rendelkező szekvenciális átalakítók periodikusan érkező bemenő jelsorozatot a bemenettel azonos periodicitású periodikusan ismétlődő kimenő jelsorozattá alakítják át. Tipikus alkalmazási példák: soros üzemmódu egyenes kód átalakítása 1-es komplement kód, vagy 2-es komplement kód (természetesen előjelvizsgálóval együtt); Kódolt információ átalakítása valamely más kóddá (pl. Bináris/Gray konverzió); soros működésű paritás vizsgáló stb.

Azokat a szekvenciális átalakítókat, amelyek csak késleltető elemet és antivalencia (kizáró VAGY) kaput tartalmaznak, lineáris szekvenciális átalakítóknak nevezik. (Más szóhasználattal a lineáris szekvenciális átalakítók késleltető és moduló 2 összeadó elemet tartalmaznak.) A lineáris szekvenciális átalakítók matematikai jellemzése lineáris szekvenciális (sorrendi) függvények-

kel történik. A lineáris szekvenciális függvény modulo 2 összeadást és késleltetést tartalmaz. Ezen átalakítók gyakorlati alkalmazási lehetősége igen nagy pl. hiba jelző és javító kódolók ill. dekódolók, digitális szűrők, kvázi véletlen számgenerátorok stb. készíthetők segítségével. Áramköri megvalósításuk visszacsatolt léptető regiszterekkel történik.

A korlátozott terjedelembre való tekintettel néhány egyszerűbb alkalmazást mutatunk be.

4.3.1. Soros üzemi Gray/bináris konverzió

Jelöljük a Gray kód k -adik bitjét G_k -val ($k = 0, 1, \dots, n$); a bináris kód k -adik bitjét B_k -val ($k = 0, 1, \dots, n$), ahol B_k és G_k értéke 0 vagy 1; a növekvő helyértékekhez tartozzék növekvő index. Könnyen belátható, hogy:

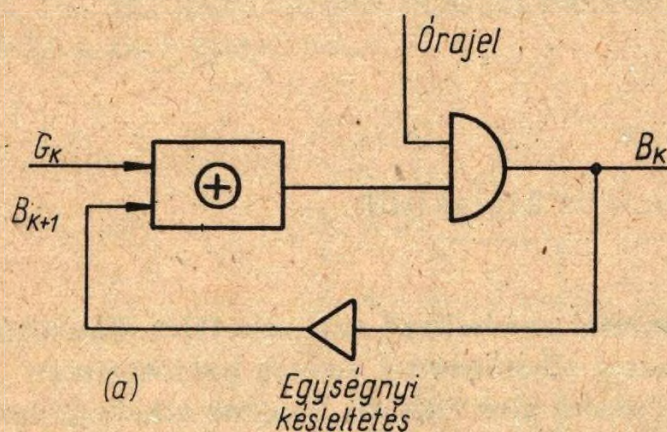
$$B_n = G_n$$

és

$$B_k = B_{k+1} \oplus G_k = \overline{B_{k+1}} G_k + B_{k+1} \overline{G_k}$$

$$k \neq n,$$

ahol tehát n mindkét kód legnagyobb helyértékéhez tartozik. (4-13/a ábra.)



G_k -t megelőző állapot		G_k -t követő állapot
B_{k+1}	G_k	B_k
0	0	0
0	1	1
1	0	1
1	1	0

(b)

4-13/a ábra

Gray-bináris konverzió soros üzemmódban

(a) A konverzió megvalósítása késleltető elem segítségével,

(b) A konverzió igazságtáblázata

A konverzióhoz szükséges késleltetés egy tároló segítségével könnyen megvalósítható, ha a B tároló G_n beérkezése előtt 0-ás állapotban van. Az igazságtáblázatot $k \neq n$ esetre nem is kellene elkészítenünk, ha figyelembe vesszük, hogy a Gray kód jegyei csökkenő indexek sorrendjében érkeznek, tehát G_{k+1} megelőzi G_k -t, így B_{k+1} is megelőzi B_k -t. (Tehát B_k jelenti a tároló állapotát a G_k -t követő ütemben, ha az előzőleg B_{k+1} állapotban volt.) Ha tehát a B_k -t a G_k beérkezése után olvassuk ki, a tároló kimenetéről a soros bináris kódba konvertált jelet vehetjük le. A G_k beérkezése utáni kiolvasás az órajel késleltetésével történhet.

A fentieknek megfelelően a B tároló logikai differencia egyenletében a C_{1B} ill. C_{2B} vezérlési függvények:

$$C_{1B} = \overline{G_k} \quad \text{és} \quad C_{2B} = G_k$$

Ha a B tároló T típusú flip-flop, akkor a bemeneti vezérlés $\overline{C_{1B}} = C_{2B}$ figyelembevételével:

$$T_B = \overline{C_{1B}} = G_k$$

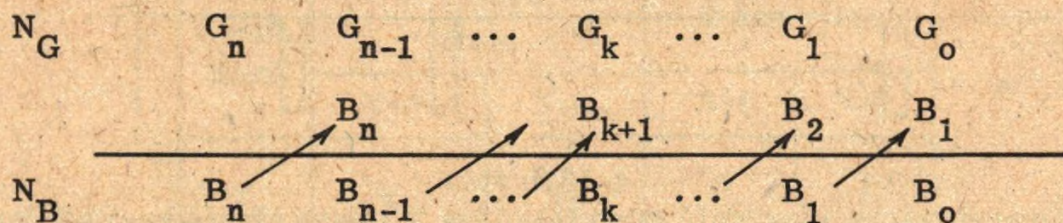
vagyis maga a soros Gray kód.

Ha a B tároló JK típusú flip-flop, akkor a bemenetek vezérlése:

$$J_B = C_{2B} = G_k \quad \text{és} \quad K_B = \overline{C_{1B}} = G_k$$

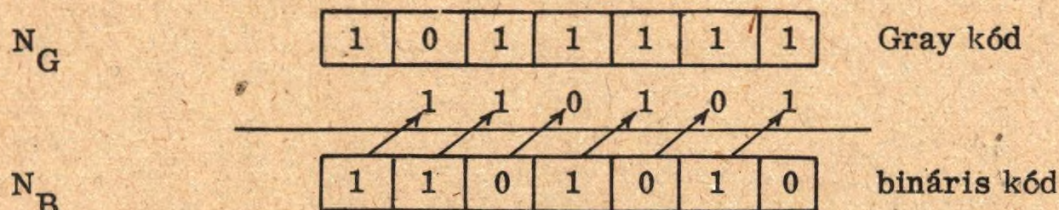
A soros üzemi Gray/bináris átalakító logikai hatásvázlatát és jelalakjait 4-13/b ábra mutatja.

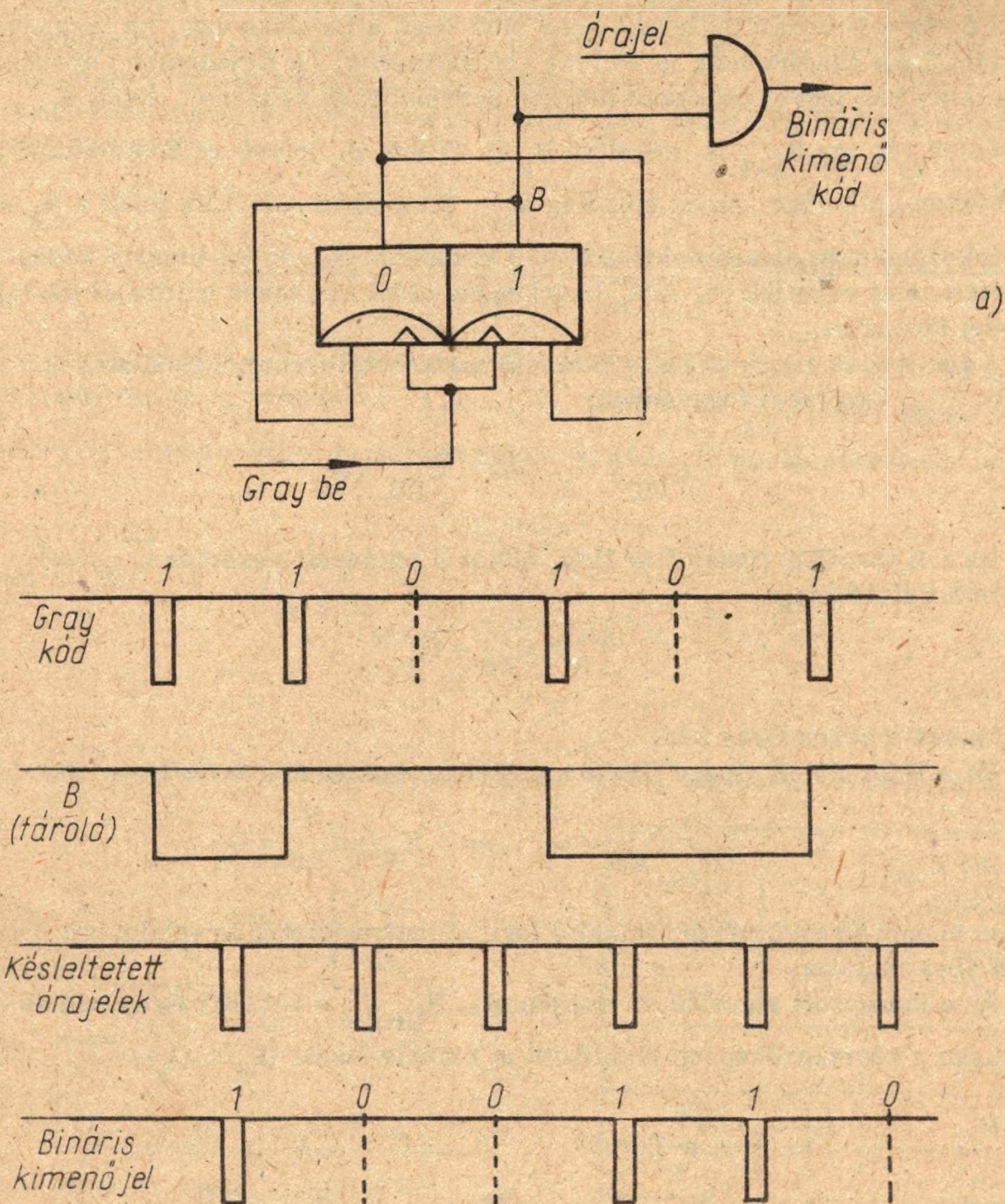
Az alkalmazott algoritmus alapján pl. $N_{GRAY} = 1011111$ Gray kód numerikusan a következőképpen alakítható át bináris kóddá ($B_n = G_n$):



ahol

$$B_k = B_{k+1} \oplus G_k$$





4-13/b ábra
 (a) Gray-bináris konverzió megvalósítása T típusu tároló segítségével
 (b) Jelalakok

4.3.2. Soros üzemű kettes komplementképző

Soros üzemmódban a kettes komplement kód többféleképpen előállítható egyenes kódból. Egyszerű algoritmus a kettes komplement előállítására a következő: a legkisebb helyértéktől a nagyobb helyértékek felé haladva a soros kód első 1-es számjegyét változatlanul hagyjuk, az összes többi számjegyben $1 \rightarrow 0$, $0 \rightarrow 1$ cserét alkalmazunk. (A választott algoritmus alkalmazása során tehát a legalacsonyabb helyértékkel kell kezdeni az átalakítást, még az előző pontban szereplő Gray/bináris konvertert először a legmagasabb helyértékkel kell működtetnünk.)

Egyszerűség kedvéért az áramkört előjelvizsgáló nélkül tervezzük meg.

A logikai tervezést ilyen egyszerű feladatnál közvetlenül az igazságtáblázat meghatározásával kezdhetjük. A bemenő változó (X) egyértelműen nem határozza meg a kimenetet (Y), ezért szekunder változót kell felvenni. Közvetlenül belátható, hogy a rendszernek két belső állapota van. A q_0 belső állapotban van a rendszer az első 1-es beérkezéséig, ebben az esetben a kimenőjel $Y=X$. Az első 1-es jelet követő valamennyi impulzusra: $Y=\bar{X}$, amit a q_1 belső állapot jellemez. A két belső állapotot egy szekunder változóval realizálhatjuk. Tehát a kombinációs hálózatot kiegészítő regiszter egyetlen tárolóból áll, ennek két állapota valósítja meg a q_0 és q_1 belső állapotot. Az állapotok kódolása a következőképpen történhet.

Belső állapot	Tároló állapot
q_0	$Q = 0$
q_1	$Q = 1$

A komplementképző állapot táblázata:

k-adik ütem	(k+1)-edik ütem		Y kimenet	
	X=0	X=1	X=0	X=1
q_0	q_0	q_1	0	1
q_1	q_1	q_1	1	0

A kódolt állapot változóval (szekunder változóval) megszerkesztett igazságtáblázat:

index	k		k+1	kimenet
	X	Q^k	Q^{k+1}	Y
0	0	0	0	0
1	0	1	1	1
2	1	0	1	1
3	1	1	1	0

A független változó: X, Q^k . A Q tároló állapot egyenlete:

$$Q^{k+1} = Q^k + X$$

Ezt a $Q^{k+1} = C_1 Q^k + C_2$ alakú általános állapot egyenlettel összehasonlítva:

$$\overline{C_1} C_2 = 0; \quad C_1 = 1; \quad C_2 = X$$

A Q tárolót pl. RS típusu tárolóval realizálva:

$$G_0 = \overline{C_1} = 0; \quad G_1 = C_2 = X$$

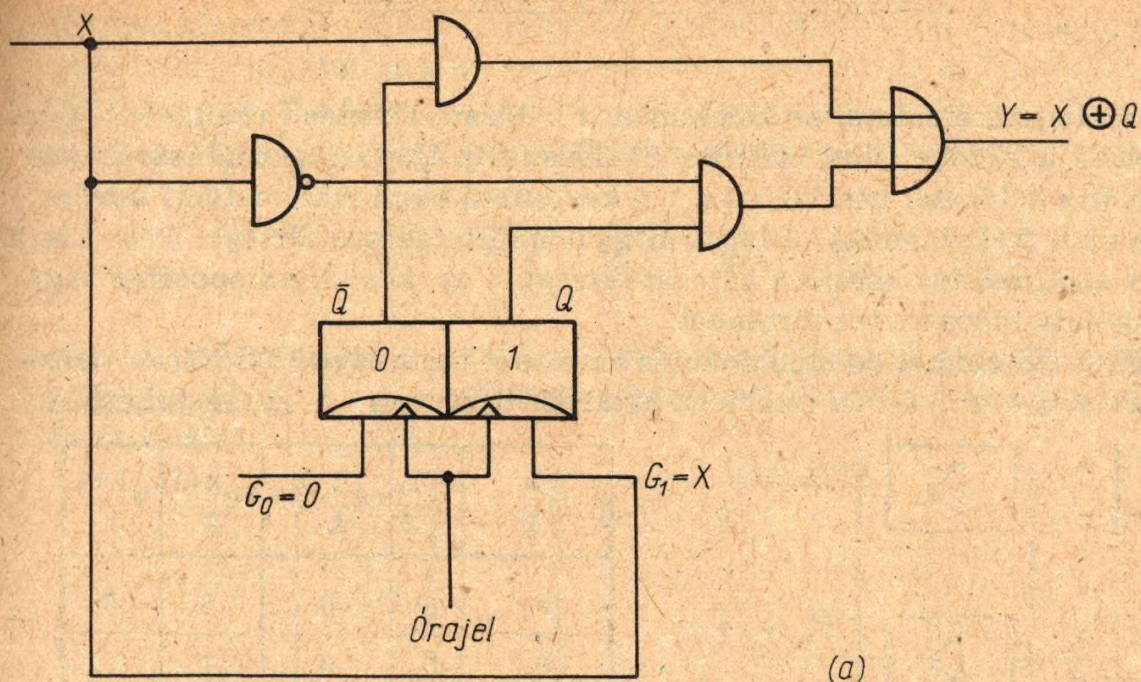
A kimenő változó: $Y = X \oplus Q^k$. A komplementképző logikai hatásvázlatát 4-14. ábrán rajzoltuk meg. Az áramkör tervezésekor feltételeztük, hogy az első impulzus beérkezése előtt a Q tároló 0-ás állapotban van.

Amennyiben a belső állapotok megvalósítására T típusu tárolót alkalmazunk, a bemenet vezérlése ($\overline{C_1} \neq C_2$)

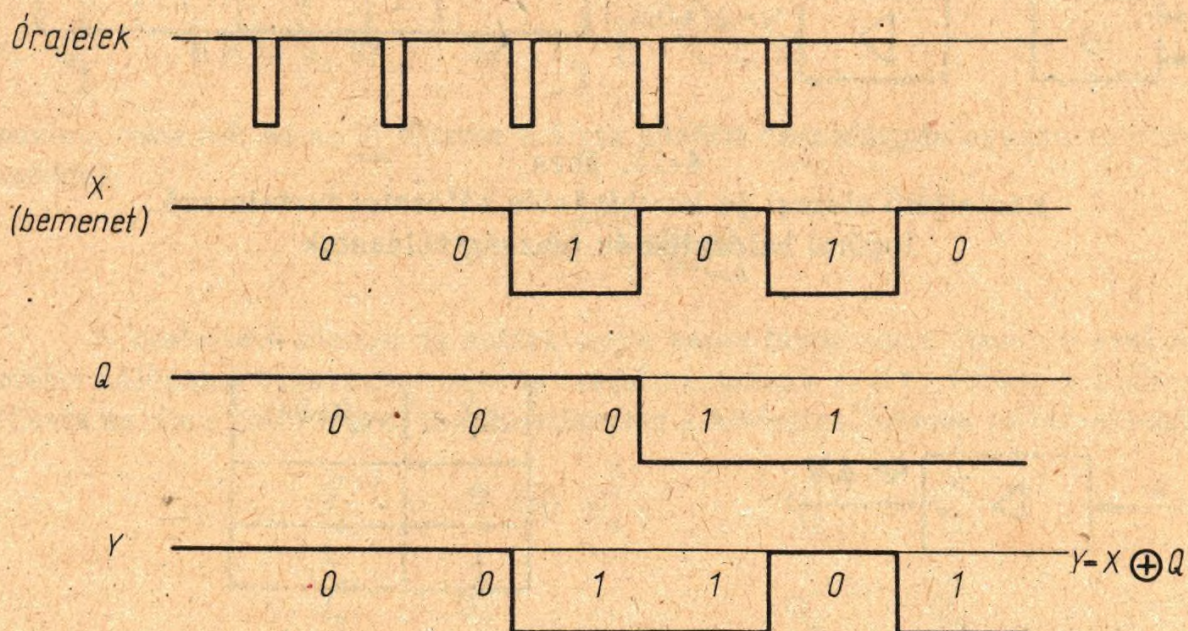
$$T_Q = \overline{C_1} Q^k + C_2 \overline{Q^k} = X Q^k$$

A vezérlési feltétel fizikailag közvetlenül belátható, a 0-ás állapotban levő tárolót az első 1-es bemenő impulzus 1-be billenti.

Még megjegyezzük, hogy miután a G_0, G_1 bemenetek vezérlése a Q szekunder változótól független, vagyis a tároló kimenete nincs hatással a statikus bemenetekre, a statikus kapuzású, dinamikus vezérlésű RS tárolón kívül a feladat aszinkron működésű statikus tárolóval is megvalósítható.



(a)



(b)

4-14. ábra

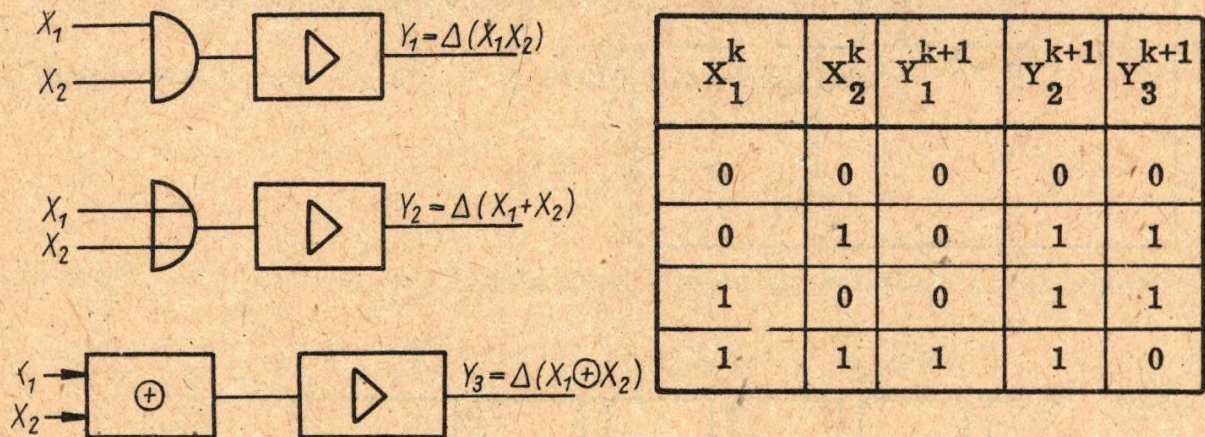
Kettes komplementképző előjelvizsgáló nélkül.

(a) Logikai hatásvázlat (b) Idődiagram

4.3.3. Lineáris szekvenciális hálózatok

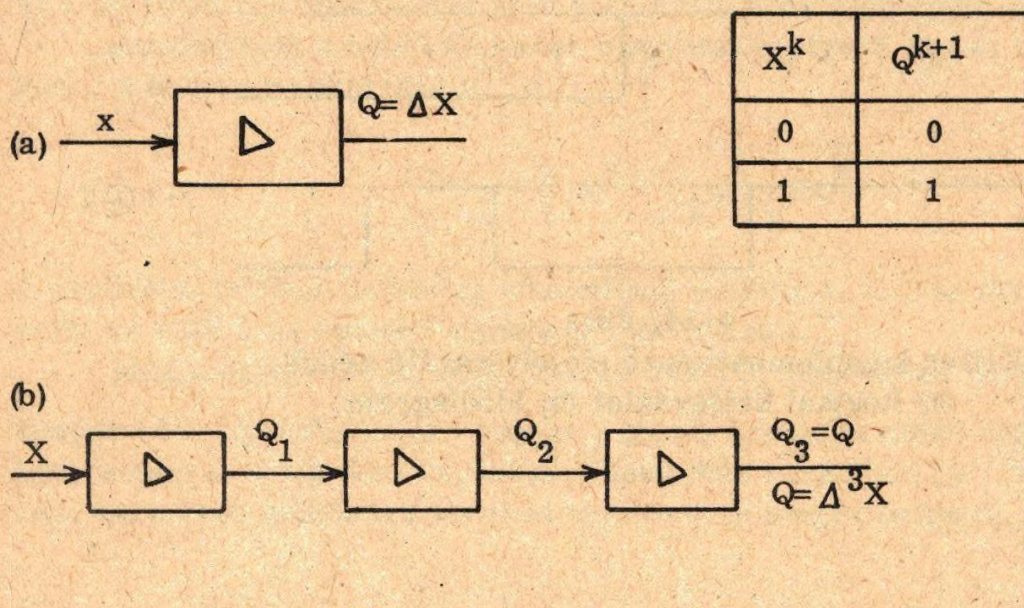
A késleltető elemmel és kombinációs hálózattal történő realizálás lehetőségére eddig már több ízben utaltunk (D típusu flip-flop felhasználásával több szekvenciális hálózatot realizáltunk.) A késleltető elem matematikai leírása eddig a k és a $(k+1)$ ütembeli állapot megkülönböztetésével történt. Több késleltető elem alkalmazása esetén a hálózat leírása a Δ késleltető operátor vagy a vektormátrix módszerrel történhet.

Késleltető elemet és kombinációs hálózatot tartalmazó hálózatok igazságtáblázatát a 4-15. ábrán láthatóan határozhatjuk meg. A Δ késleltető



4-15. ábra

Késleltető elemet és kombinációs hálózatot tartalmazó logikai hálózatok és igazságtáblázatuk



4-16. ábra

Több késleltető elem láncbakapcsolása

operátort a 4-16. ábra igazságtáblázata alapján értelmezzük. A három késleltető elem láncbakapcsolása, amint ezt már megterveztük, egy léptető regiszternek felel meg, amely a bemenetére adott jelet három egységnyi (3 bit) késleltetéssel adja ki. A kimenő jel: $Q = \Delta^3 X$ vagy más alakban: $Q^{k+3} = X^k$. A $Q = Q_1, Q_2, Q_3$ belső változók között a következő összefüggés van:

$$Q_1^{k+1} = X^k, \quad Q_2^{k+1} = Q_1^k, \quad Q_3^{k+1} = Q_2^k$$

A szabályozás elméletben szokásos módon az X bemenő változót is állapotváltozónak tekintve: $X = Q_0$ jelöléssel a rendszer állapot jellemzése vektor mátrix alakban:

$$\begin{bmatrix} Q_1^{k+1} \\ Q_2^{k+1} \\ Q_3^{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Q_0^k \\ Q_1^k \\ Q_2^k \end{bmatrix}$$

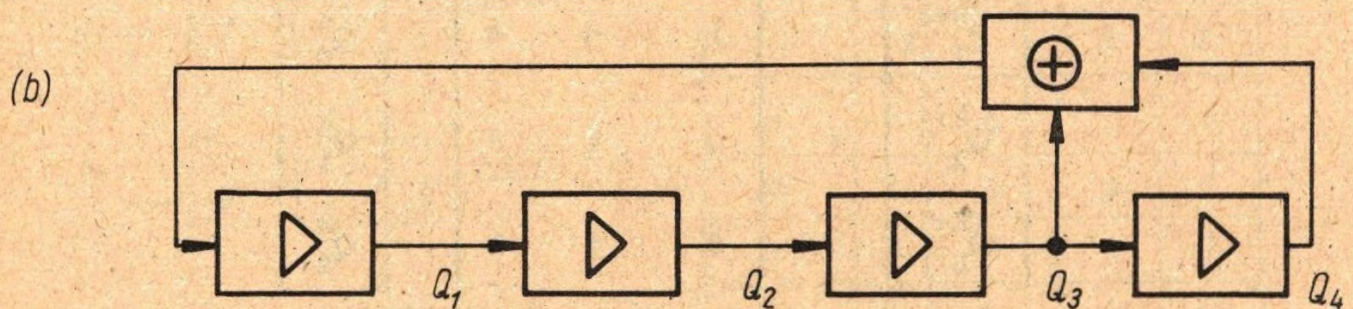
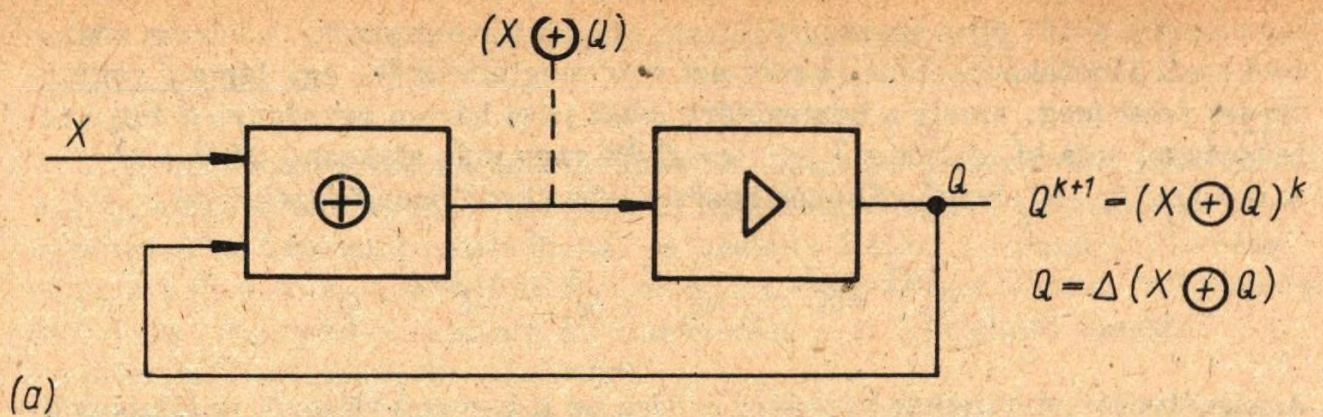
$$A \quad \underline{Q_i}^{k+1} \quad (i = 1, 2, 3) \quad \text{és} \quad \underline{Q_j}^k \quad (j = 0, 1, 2)$$

oszlop vektorok és az M állapot mátrix közötti összefüggés egyszerűbb írásmóddal:

$$\underline{Q_i}^{k+1} = \underline{M} \underline{Q_j}^k$$

A késleltető elemet és antivalencia kaput tartalmazó lineáris szekvenciális hálózatok visszacsatolást is tartalmazhatnak (4-17. ábra). A b ábra hálózata az ábra jelölésével matematikailag a következőképpen jellemezhető:

$$\begin{aligned} Q_1^{k+1} &= Q_3^k \oplus Q_4^k, & Q_2^{k+1} &= Q_1^k \\ Q_3^{k+1} &= Q_2^k, & Q_4^{k+1} &= Q_3^k \end{aligned}$$



4-17. ábra
 Lineáris szekvenciális hálózatok

A Q állapot változók k és $(k+1)$ ütemben felvett értéke közötti kapcsolat vektor-mátrix alakban:

$$\begin{bmatrix} Q_1^{k+1} \\ Q_2^{k+1} \\ Q_3^{k+1} \\ Q_4^{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Q_1^k \\ Q_2^k \\ Q_3^k \\ Q_4^k \end{bmatrix}$$

ahol a mátrix szorzás összeadása moduló 2 szerinti összeadást jelent. Ha a rendszer valamely Q^0 oszlop vektorral jellemezett kezdeti állapotból indul, akkor az egymást követő órajelek hatására a következő állapotokat veszi fel:

$$\underline{Q}^0, \underline{M}\underline{Q}^0, \underline{M}^2\underline{Q}^0, \underline{M}^3\underline{Q}^0, \dots, \underline{M}^n\underline{Q}^0$$

Amennyiben az $\underline{M}$ állapot mátrix nem szinguláris (vagyis az $|M|$ determinánsa

nem zérus), akkor egy adott állapotot létrehozó (megelőző) Q állapot vektor is meghatározható:

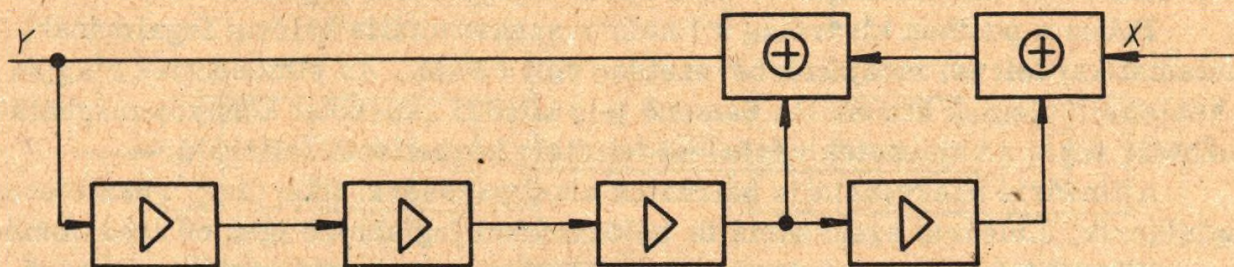
$$\underline{Q}^k = \underline{M}^{-1} \underline{Q}^{k+1}, \text{ ha } \underline{M}^{-1} \text{ létezik.}$$

Pl. jelen esetben:

$$\underline{M}^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Amennyiben a vizsgált visszacsatolt hálózatot még egy antivalencia kapuval kiegészítve a bemenő jelet (X) is feltüntetjük (4-18. ábra), könnyen beláthatóan a kimenet (Y) a következő lineáris sorrendi függvénnyel írható fel:

$$Y \oplus \Delta^3 Y \oplus \Delta^4 Y = X$$



4-18. ábra

Lineáris szekvenciális áramkör visszacsatolással

A lineáris sorrendi függvény linearitását felhasználva értelmezhető a kimenet-bemenet között a következő szekvenciális (digitális) átviteli függvény (figyelembe véve, hogy ha $C = A \oplus B$, akkor $B = A \oplus C$):

$$\frac{Y}{X} = \frac{1}{1 + \Delta^3 + \Delta^4}$$

(Láthatóan a szekvenciális átviteli függvény és a mintavételezett hálózatok impulzus átviteli függvénye között könnyen lehet analógias kapcsolatot teremteni.)

Amennyiben a bemenő jel zérus, és létezik egy olyan Y bináris sorozat, amelyre:

$$Y \oplus \Delta^3 Y \oplus \Delta^4 Y = 0$$

akkor a kimeneten jelet (bináris sorozatot) kapunk akkor is, ha nincs bemenő jelsorozat. Hasonló eredményre juthatunk az $\underline{\underline{M}}$ mátrix $\Phi(x)$

$$\Phi(x) = | \underline{\underline{M}} - x \underline{\underline{I}} |$$

karakterisztikus polinomjának meghatározásával is. Esetünkben, a kivonás helyett a modulo 2 összeadást alkalmazva:

$$\left| \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \oplus \begin{bmatrix} x & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x \end{bmatrix} \right| = 1 \oplus x^3 \oplus x^4$$

A karakterisztikus egyenlet:

$$\Phi(\underline{\underline{M}}) = \underline{\underline{I}} \oplus \underline{\underline{M}}^3 \oplus \underline{\underline{M}}^4$$

A zérus bemenő sorozathoz tartozó periodikus Y kimenő sorozat meghatározásával, tulajdonságaival nem ebben a jegyzetben foglalkozunk.

Ebben a pontban kizárólag a lineáris szekvenciális hálózat fogalmának, matematikai leírási módjának bevezetése volt célunk. Az elmondottak alapján a késleltető elemek kimenő és bemenő jele közötti kapcsolat könnyen meghatározható, tehát a bemenetek vezérlési feltétele közvetlenül felírható.

A lineáris szekvenciális hálózatok alkalmazásával több tárgy részletesen foglalkozik. Lineáris szekvenciális hálózatokkal foglalkozó igen bő irodalomból kiemeljük a hálózatok analízise terén HUFFMAN, véges automatákra történő alkalmazási munkákból SOLOMON, hibajavító kódolásra történő alkalmazási munkákból pedig PETERSON eredményeit.

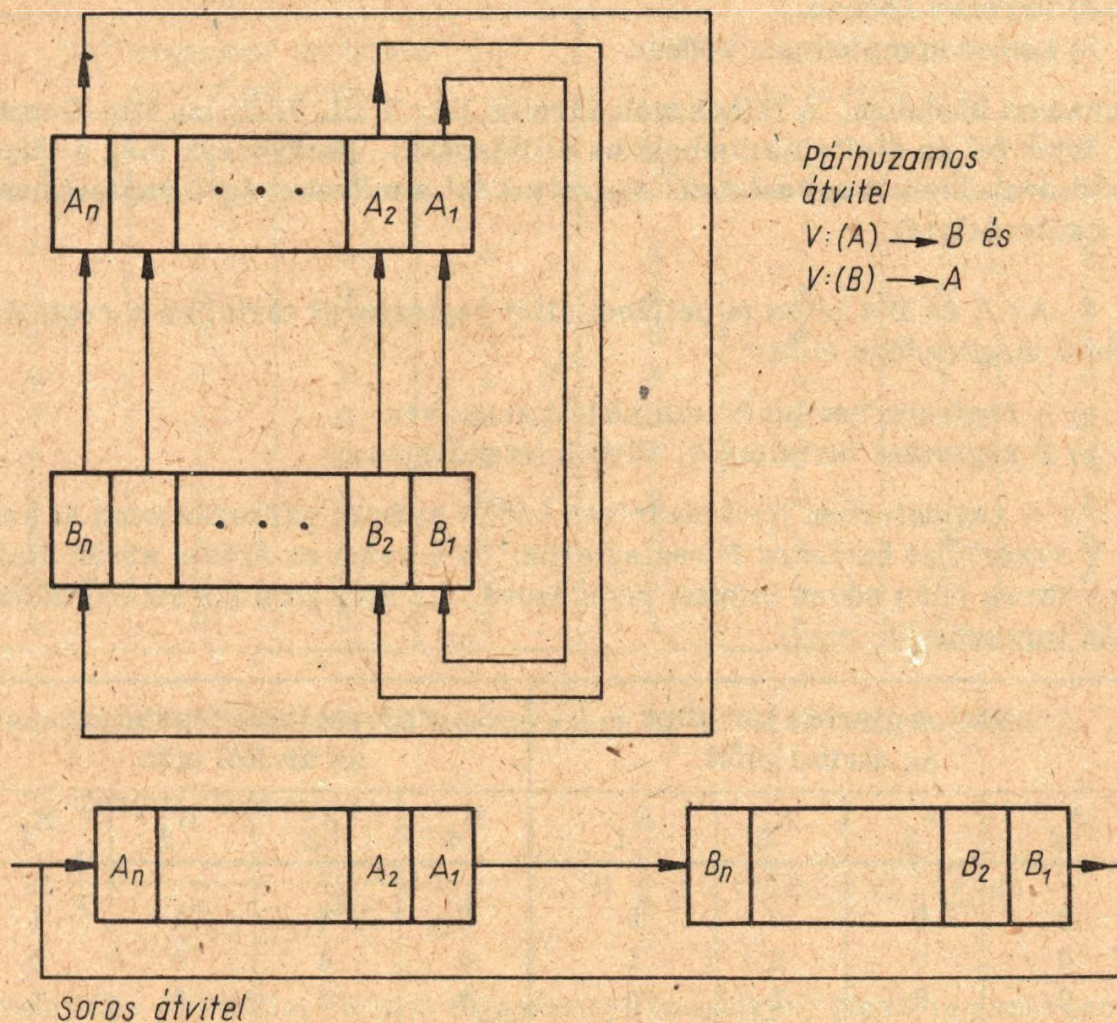
4.3.4. Gyakorló feladatok

1. Tervezendő statikus vezérlésű aszinkron RS tároló felhasználásával léptető regiszter, mely alkalmas:

- Párhuzamos kódban adott információ beírására (V_B vezérlőjel hatására)
- V_T vezérlőjel hatására a tárolt információ törlésére
- V_J vezérlőjel hatására a regiszter tartalmát egy helyértékkal jobbra lépteti
- V_B jel hatására a regiszter tartalmát egy helyértékkal balra lépteti.

Készítsük el a léptető regiszter igazságtáblázatát, írjuk fel az állapot-egyenletét, valamint a tárolók vezérlésének bemeneti egyenleteit.

2. Az A és B "n" cellából álló regiszter tartalmát egyidejűleg át kell vinni egymásba (4-19. ábra): $V: (A) \rightarrow B$, $V: (B) \rightarrow A$



4-19. ábra

A regiszterek RS, T, ill. JK típusú tárolókból állnak. Határozzuk meg a tárolók bemenetének vezérlésére szolgáló logikai egyenleteket, ha az átvitel

- soros üzemmódban
- párhuzamos üzemmódban történik, rajzoljuk fel a regiszterek logikai hatásvázlatát.

3. V vezérlőjel hatására A regiszter tartalma átírandó B regiszterbe, de átírás közben a szám fordított sorrendben rendeződik. Az átvitel párhuzamos üzemmódban történik. A regisztereket RS, ill. T típusú tárolókkal valósítjuk meg.

Írjuk fel az átvitelt, szimbolikus jelöléssel. Határozzuk meg a tárolók vezérlésének logikai egyenleteit.

4. Az A regiszter tartalma V vezérlőjel hatására átiródik a B regiszterbe, mégpedig az M műveletvezérlő tároló tartalmától függően:

- a) egyenes kódban
- b) kettes komplementum kódban

párhuzamos üzemben. A felhasznált tárolók RS, T ill. D típusu flip-floppok.

Írjuk fel az átvitelt szimbolikus állításokkal. Határozzuk meg a tárolók vezérlésének logikai egyenleteit. Rajzoljuk fel a műveletvégző regiszterek logikai hatásvázlatát.

5. Az A és B 4 bites (4 cellából álló) regiszterek tartalma a regiszter operáció megkezdése előtt:

- a) A regiszterben binárisan kódolt szám van
- b) B regiszter tartalma 0, illetve tetszőleges.

Az A regiszterben levő binárisan kódolt számot párhuzamosan át kell vinni V vezérlőjel hatására B regiszterbe, de a szám az átvitel során átalakul excess-three (más néven Stibitz) kódolásúvá. Az átvitelt a következő táblázat alapján tervezhetjük meg:

"A" regiszter tartalma az átvitel előtt				"B" regiszter tartalma az átvitel után			
A ₄	A ₃	A ₂	A ₁	B ₄	B ₃	B ₂	B ₁
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Rajzoljuk fel a logikai hálózat állapot diagramját. Írjuk fel az átvitelt szimbolikus állításokkal. Határozzuk meg a tárolók bemenetének vezérlésére szolgáló logikai egyenleteket, ha a tárolók

- a) RS illetve inverz RS
- b) T illetve inverz T

típusúak.

6. Az A és B 4 cellából álló regiszter. A regiszter tartalmát az előző feladatban leírt módon át kell vinni a B regiszterbe párhuzamosan. Az átvitt szám az átvitel során Gray kódban rendeződik.

"A" regiszter tartalma az átvitel előtt				"B" regiszter tartalma az átvitel után			
A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

A feladat egyéb követelményei megegyeznek az 5. feladatban leírtakkal.

7. Tervezzük meg az

$$a) \frac{Y}{X} = \frac{1}{1 \oplus \Delta \oplus \Delta^3 \oplus \Delta^4}$$

$$b) \frac{Y}{X} = 1 \oplus \Delta \oplus \Delta^3 \oplus \Delta^4$$

szekvenciális (digitális) átviteli függvénnyel rendelkező lineáris szekvenciális hálózatokat. Rajzoljuk fel a hálózatok logikai hatásvázlatát.

8. Határozzuk meg a következő sorrendi függvények kimenő jelsorozatát:

$$a) Q = \Delta \bar{Q}$$

$$b) Q = \Delta^2 \bar{Q}$$

$$c) Q = \Delta (\bar{Q} \Delta \bar{Q}) = (\Delta \bar{Q}) (\Delta^2 \bar{Q})$$

ha a hálózatok bemenetét az órajelek vezérlik (az órajelet "1" generátornak tekintve a sorrendi függvényekben nem tüntettük fel).

Megoldás: A hálózatot vezérlő órajelek beérkezésének időpontját jelöljük $t_1, t_2, t_3, t_4, \dots$ -vel. Az egyes hálózatok kimenő jelsorozata:

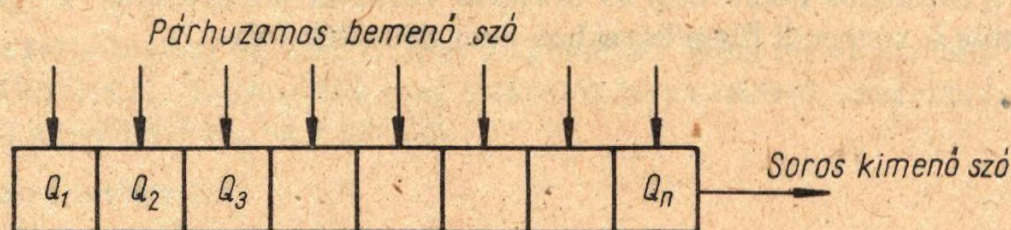
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	$\dots$
Órajel	1	1	1	1	1	1	1	1	$\dots$
$Q = \Delta \bar{Q}$	0	1	0	1	0	1	0	1	$\dots$
$Q = \Delta^2 \bar{Q}$	0	0	1	1	0	0	1	1	$\dots$
$Q = \Delta(\bar{Q} \Delta \bar{Q})$	0	0	1	0	0	1	0	0	$\dots$

A táblázat kitöltésénél feltételeztük, hogy a hálózat a " t_1 " időpont előtt 0 állapotban van.

Rajzoljuk fel az egyes sorrendi hálózatok logikai hatásvázlatát.

4.4 Üzem mód átalakítók

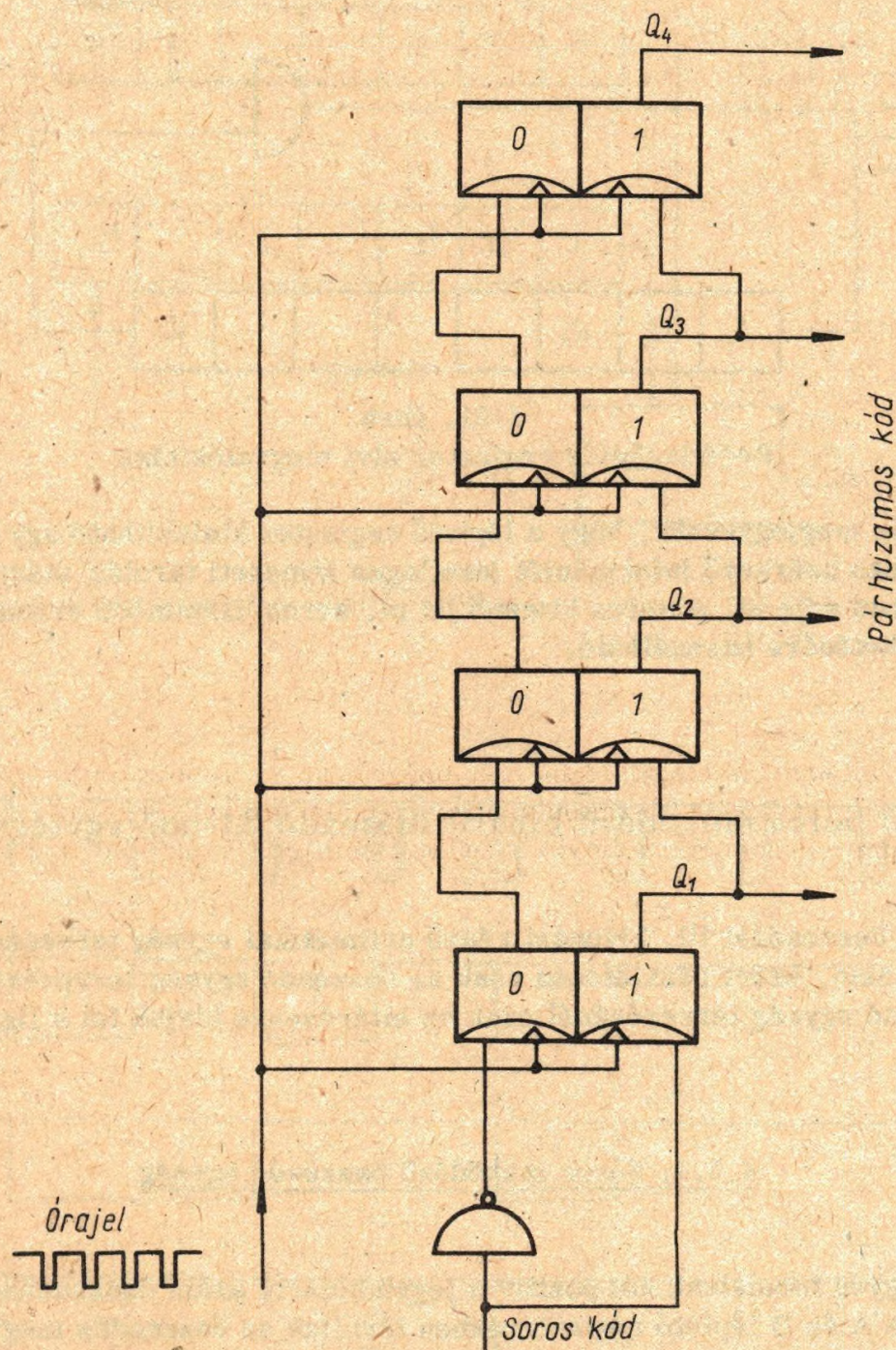
Gyakori feladat soros (párhuzamos) kódban adott bemenő információ átalakítása párhuzamosan (sorosan) kódolt kimenő jellé. Az átalakítást elvégző.



4-20/a ábra

(a) Soros/párhuzamos (b) Párhuzamos/soros üzemmódváltó elvi megvalósítása

üzemmód váltókat soros-párhuzamos ill. párhuzamos-soros átalakítóknak nevezzük. Megvalósításuk legegyszerűbben léptető regiszterek segítségével történhet (4-20. ábra). A soros-párhuzamos átalakító a soros kód egymás után érkező helyértékeit a léptető regiszter első fokozatába írja be, az órajelek hatására pedig lépteti. A teljes kódszó beérkezése után abból párhuzamosan kiolvasható. A párhuzamos soros átalakító a párhuzamosan beírt információt kilépteti a regiszterből, tehát a regiszter valamelyik szélső fokozatáról soros

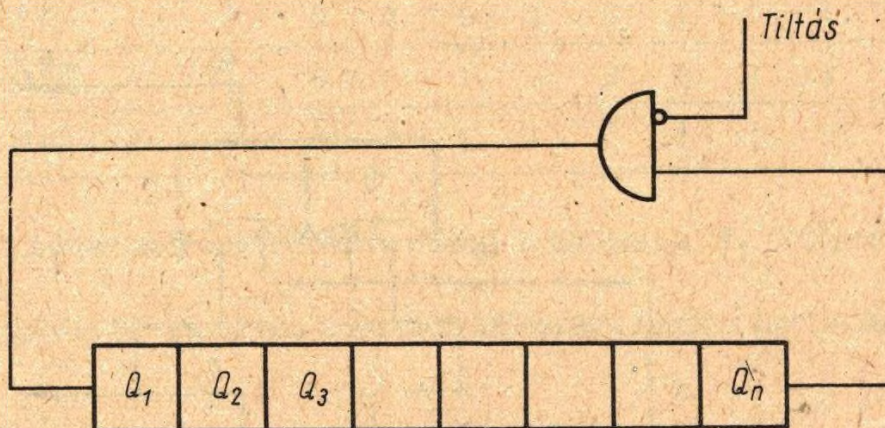


4-20/b ábra

Soros/párhuzamos üzemmódváltó logikai hatásvázlata

kódban vehető le. A soros párhuzamos üzemmód átalakító logikai hatásvázlatát a 4-20. ábra mutatja.

Recirkulációs regiszter készíthető a léptető regiszterből, ha az utolsó billenőkört az elsővel kötjük össze (4-21. ábra). A kialakított zárt hurokban az információ mindaddig kering, amíg a visszacsatoló ágba helyezett inhibítáló kapuval le nem tiltjuk. Ilyen áramkörre akkor van szükségünk, ha a léptetés után az információt más célra fel akarjuk használni.



4-21. ábra

Recirkulációs regiszter elvi megvalósítása

Még megjegyezzük, hogy a léptető regiszter kialakítható úgy is, hogy a soros kódban beérkező információt (esetleges átmeneti tárolás után) ugyancsak soros kódban adja ki. A soros kimenő jel pl. soros üzemmódu aritmetikai egység működtetésére használható.

4.5 Soros működésű bináris összeadó, kivonó egységek

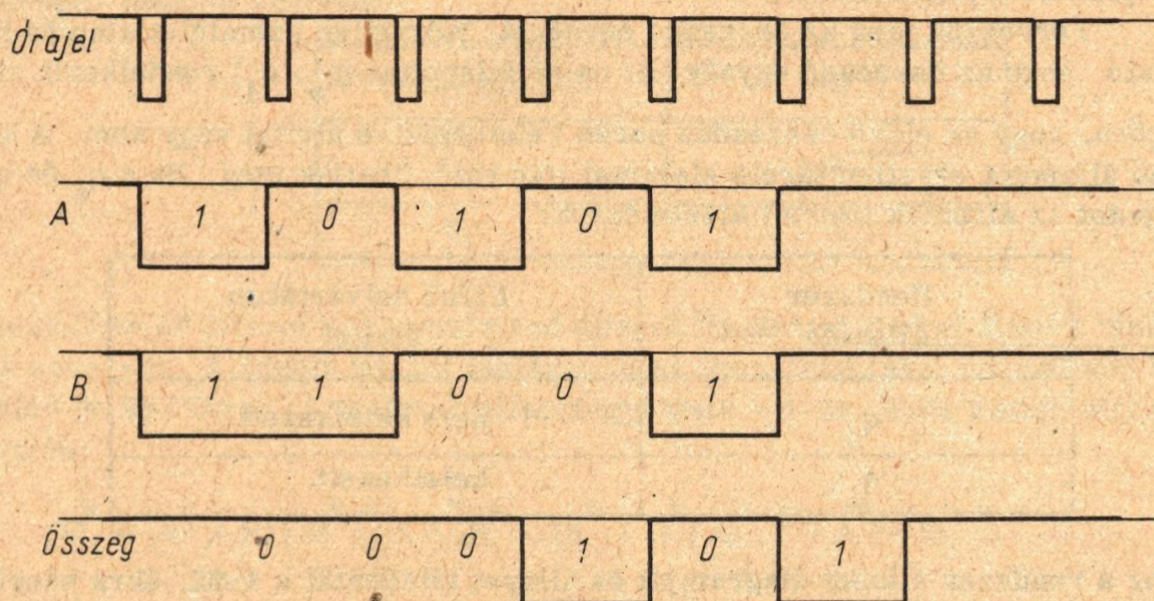
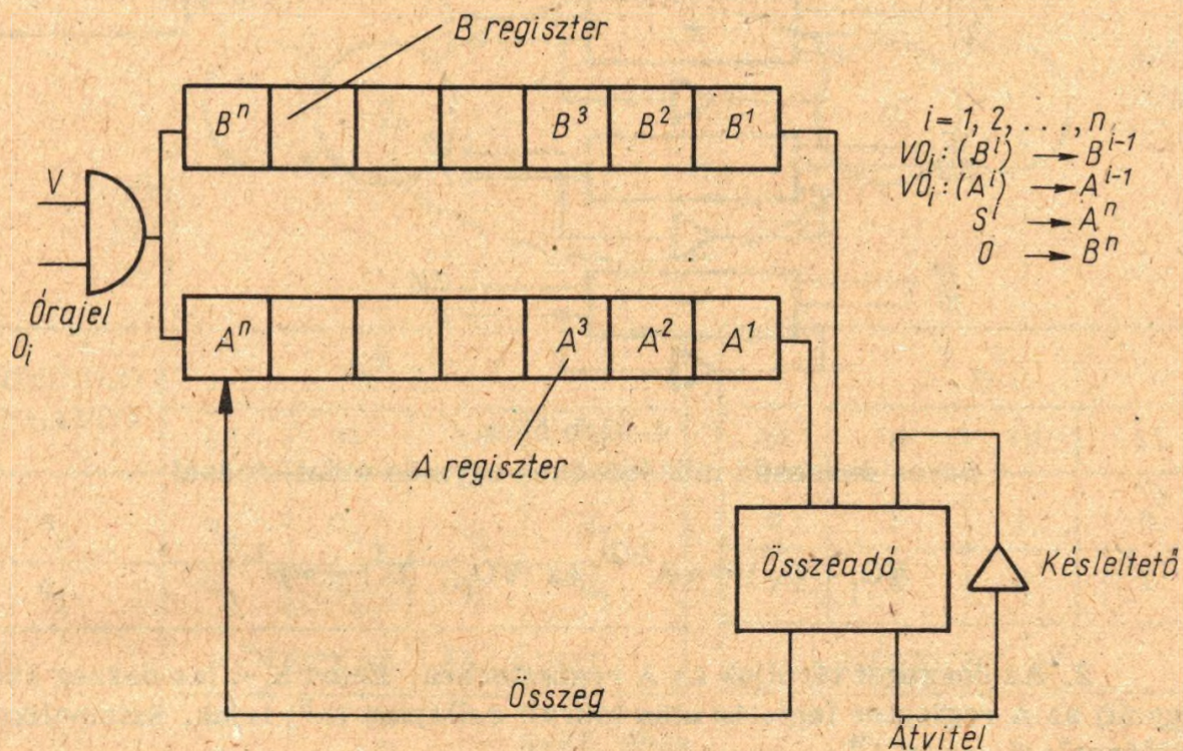
Az összeadást ill. kivonást végző aritmetikai egység tervezése hasonlóan végezhető, ezért részletesen csak az összeadó egység tervezését mutatjuk be, a kivonó egység tervezésénél csak az eltérésekre hívjuk fel a figyelmet.

4.5.1. Soros működésű összeadó egység

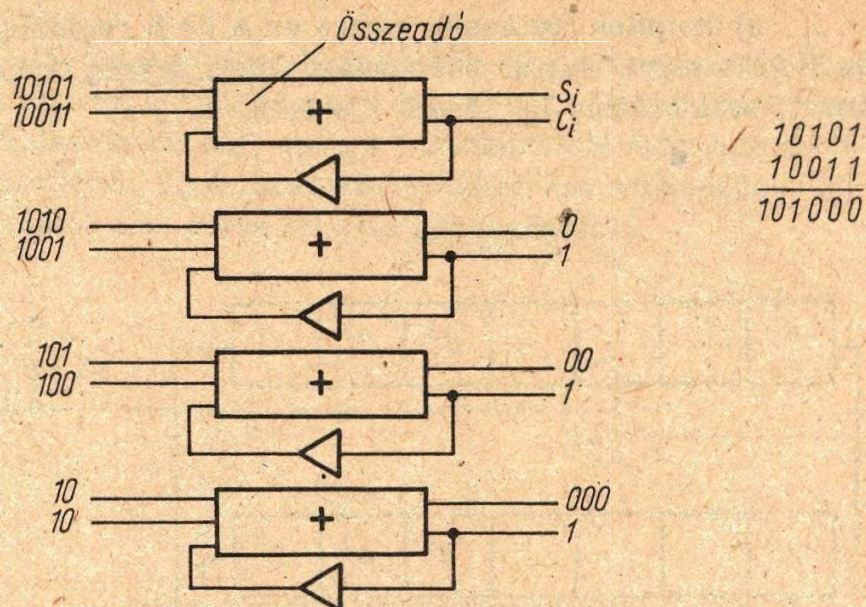
Legyen feladatunk két pozitív n jegyű bináris szám összeadása. A számjegyeket az A és B léptető regiszterekben tároljuk az összeadás megkezdése előtt. Az összeget tároljuk az A (akkumulátor) regiszterben.

Az összeadás során a következő feladatokat kell elvégezni.

1. Az összeadás kezdetét jelentő V vezérlőjel hatására az 0_i ($i = 1, 2, \dots, n$) órajelek léptessék jobbra az A és B regiszter tartalmát. Az összeadandó számjegyek így az összeadást végző egység bemenetére kerülnek (4-21. ábra), szimbolikus jelöléssel ($i = 1, 2, \dots, n$):



4-21/a ábra
Soros üzemű összeadó elvi vázlata és jelalakjai



4-21/b ábra

Soros összeadó működésének egymás utáni lépései

$$VO_i : (A^i) \rightarrow A^{i-1} \text{ és } VO_i : (B^i) \rightarrow B^{i-1}$$

2. Az összeget tároljuk az A regiszterben. Ezért S_i -t (az összeg számjegyeit) az A regiszter léptetés után kiürült cellájába (A^n) írjuk. Szimbolikus jelöléssel: $S_i \rightarrow A^n$.

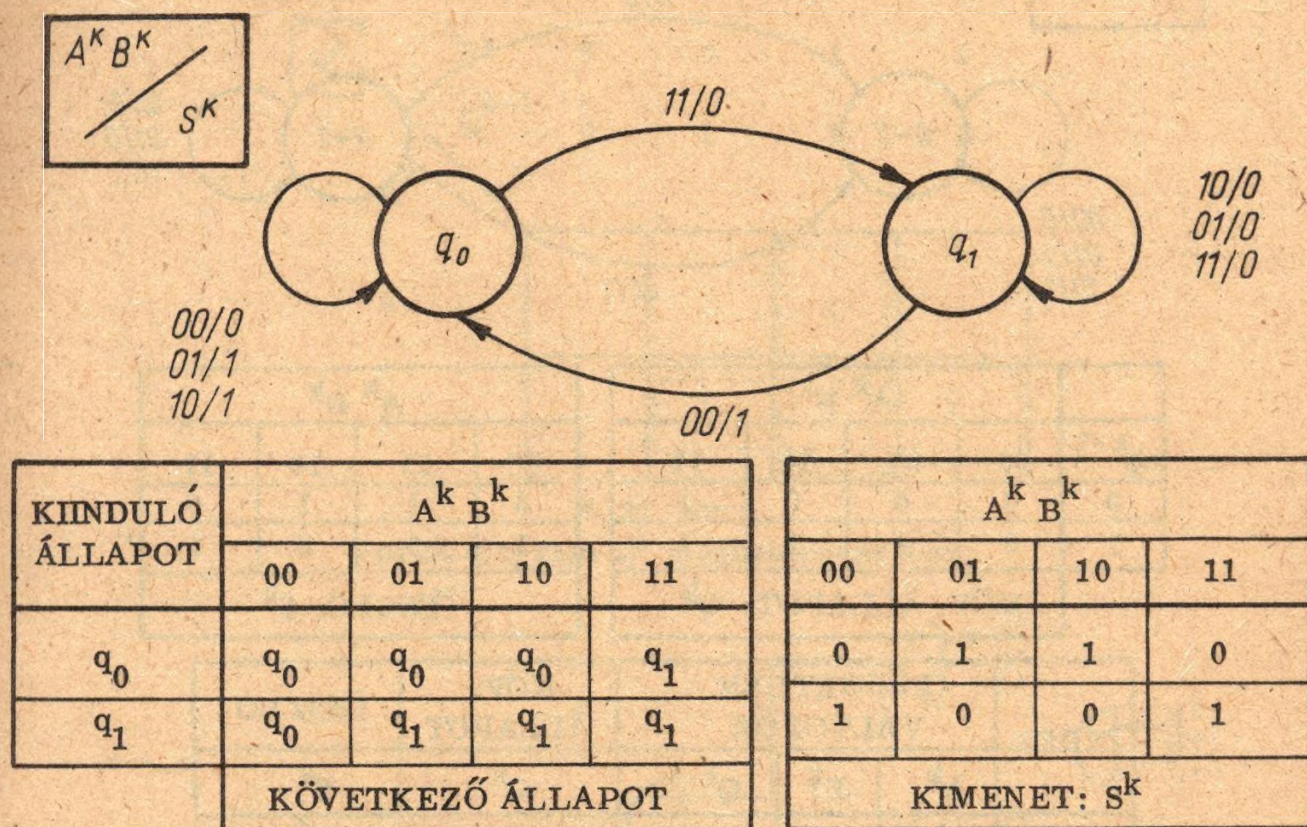
3. B regiszterbe visszairhatjuk a B számot, ha recirkulációs regiszternek képezzük ki. Egyszerűség kedvéért a B regiszter léptetés után kiürült cellájába ne írjunk semmit: $0 \rightarrow B^n$.

Tervezzük meg az összeadó egységet. Közvetlen szemlélet alapján belátható hogy az összeadó egység két belső állapottal (q_0 , q_1) rendelkezik attól függően, hogy az előző összeadás során keletkezett-e átvitel vagy nem. A két belső állapotot egyetlen tároló elemmel (Q) valósíthatjuk meg. Ha a q_0 és q_1 állapotot az alábbiak szerint értelmezzük:

Rendszer állapota	Előző helyértéken átvitel
q_0	nem keletkezett
q_1	keletkezett

akkor a rendszer állapot diagramját és állapot táblázatát a 4-22. ábra szerint határozhatjuk meg. Az állapot diagramon az A és B számnak a k-adik ütem során az összeadóba belépő A^k , B^k számjegyet és ugyancsak a k-adik ütemben

előállított összeg számjegyet (S^k) tüntettük fel. (Az összeget az előző hely-
értéken keletkező átvitel figyelembevételével határozzuk meg.)



4-22. ábra

Soros összeadó állapot diagramja és állapot táblázata

A két belső állapothoz a Q tároló két állapotát rendeljük hozzá a következőképpen (állapot kódolás):

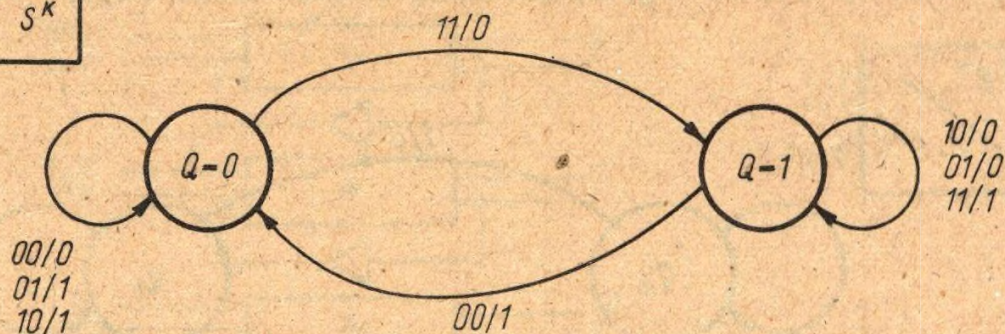
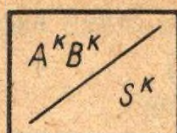
Belső állapot	Q
q_0	0
q_1	1

A kódolt belső állapotokat tartalmazó állapot táblázatot Kódolt állapot táblázatnak nevezzük. A kódolt állapot táblázatból az igazságtáblázat közvetlenül felírható (4-23. ábra). A Q tároló állapotegyenlete a 4-24. ábra Veitch diagramja alapján:

$$Q^k = (A^k + B^k) Q^{k-1} + A^k B^k = (A^k + B^k) Q^{k-1} + A^k B^k \overline{Q}^{k-1}$$

Az általános állapot egyenlet alapján a C_1 és C_2 vezérlési függvények
($\overline{C_1} C_2 = 0$)

$$C_1 = A^k + B^k, \quad C_2 = A^k B^k$$



	$A^k B^k$			
Q^{k-1}	00	01	10	11
0	0	0	0	1
1	0	1	1	1
KÖV. ÁLLAPOT: Q^k				

	$A^k B^k$			
	00	01	10	11
0	0	1	1	0
1	1	0	0	1
ÖSSZEG: S^k				

INDEX	FÜGGETLEN VÁLTOZÓK			KÖV. ÁLLAPOT	ÖSSZEG
	A^k	B^k	Q^{k-1}	Q^k	S^k
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

4-23. ábra

Összeadó egység kódolt állapot táblázata, igazságtáblázata

A Q tárolót statikus kapuzású, dinamikus vezérlésű RS tárolóval realizálva a bemeneti egyenletek:

$$G_0 = \overline{C_1} = \overline{A^k B^k}, \quad G_1 = C_2 = A^k B^k$$

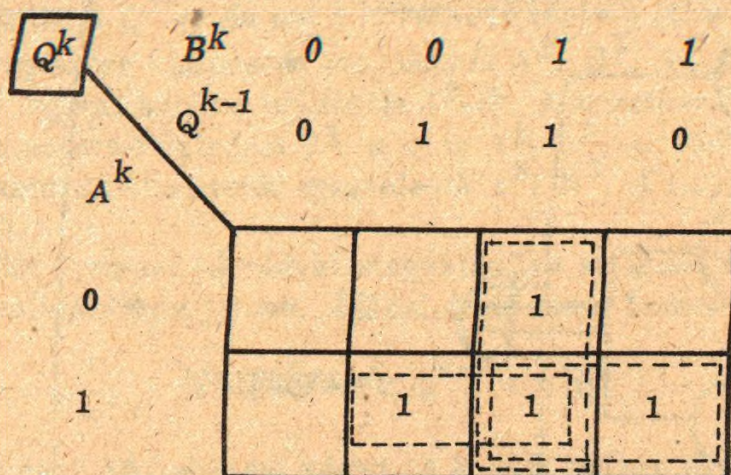
A k -adik helyértéken keletkező összeg az igazságtáblázatból kiolvasható:

$$S^k = A^k \oplus B^k \oplus Q^{k-1}$$

ill.

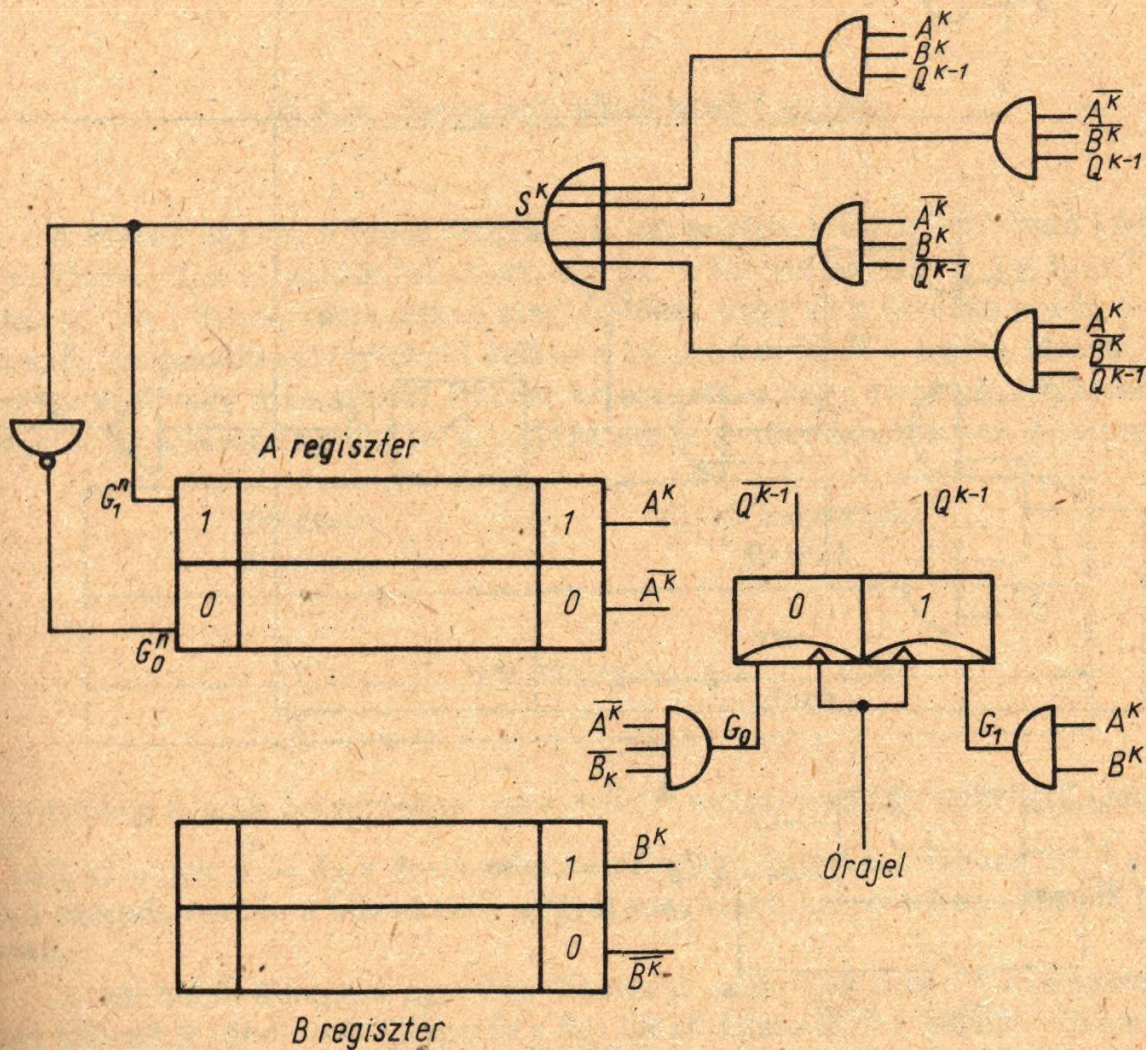
$$S^k = (A^k B^k + \overline{A^k B^k}) Q^{k-1} + (\overline{A^k B^k} + A^k B^k) \overline{Q^{k-1}}$$

Az áramkör (4-25. ábra) működése a következő. Az A és B szám első számjegye (A^1 , B^1) megjelenik az összeget előállító kapuk és a Q triggerhez kapcsolódó kapuk bemenetén. Az összeadás kezdetekor Q nullás állapotban van,



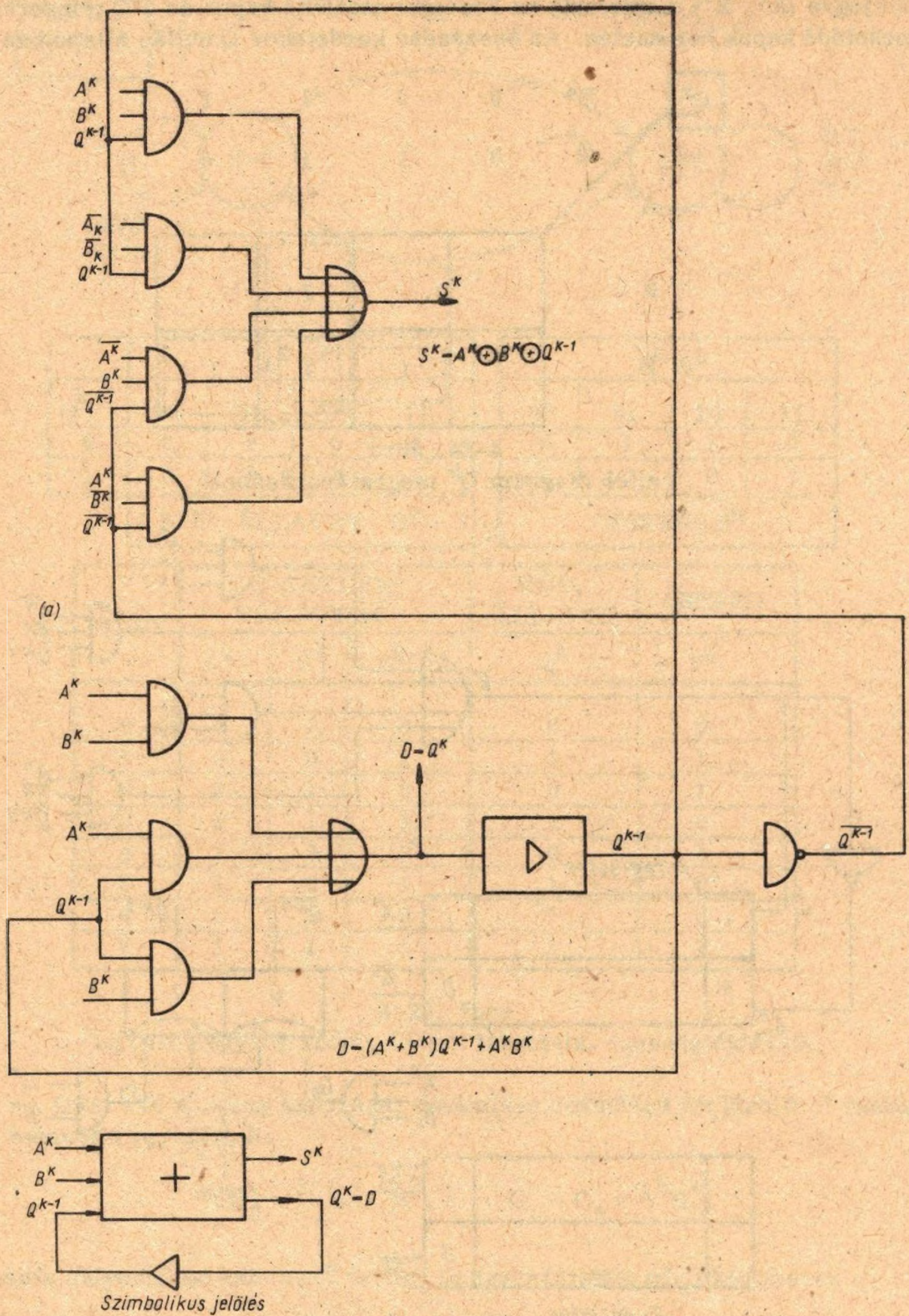
4-24. ábra

Veitch diagram Q^k meghatározásához



4-25. ábra

Soros üzemű összeadó elvi logikai hatásvázlata RS tároló alkalmazása esetén



4-26. ábra
 Soros összeadó elvi realizálása D típusú tárolóval
 (b) szimbolikus jelölés

$Q^0 = 0$. Az S^1 összeg megjelenik az eredmény regiszter kapuzó bemenetein. A léptető órajel hatására az integráló jellegű kapuzó bemenetek vezérlésétől függően S^1 beíródik az eredmény regiszterbe. A Q trigger Q^0 , A^1 , B^1 -nek megfelelően előállítja Q^1 -et és ezt a következő léptető jelig tárolja. Ugyanakkor az A és B regiszter kimenetén megjelenik A^2 , B^2 . Az A^2 , B^2 , Q^1 -ből az összeget képező logikai hálózat előállítja S^2 -öt. A következő órajel hatására az eredmény regiszterbe beíródik S^2 , a Q az A^2 , B^2 , Q^1 -ből előállítja Q^2 -t és az A, B regiszterek kimenetén megjelenik A^3 , B^3 . A folyamat hasonlóan folytatódik tovább.

Az összeadót D típusú tárolóval megvalósítva az irodalomban leggyakrabban előforduló realizációhoz jutunk. A késleltető elem bemeneti egyenlete:

$$D = (A^k + B^k) Q^{k-1} + A^k B^k$$

A D típusú tároló csak késleltetést végez, a k-adik helyértéken A^k , B^k és Q^{k-1} segítségével kombinációs hálózattal előállított átviteli impulzust (Q^k) a (k+1)-edik helyértéken veszi figyelembe az összeadó (4-26. ábra).

4.5.2. Soros működésű kivonó egység

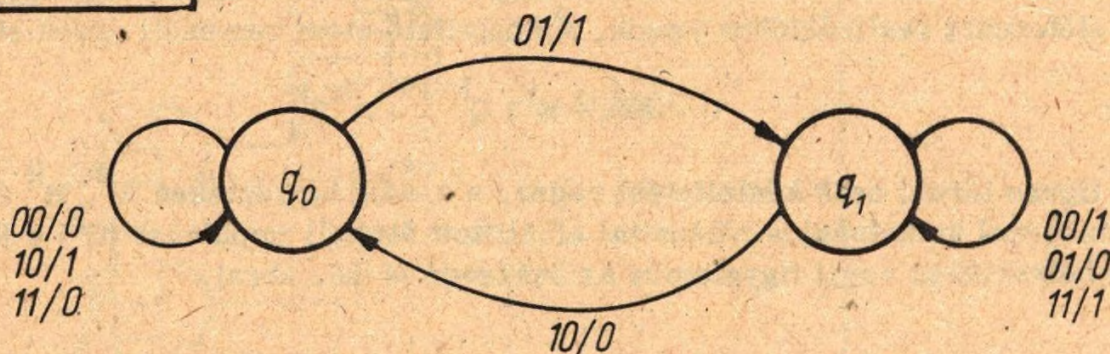
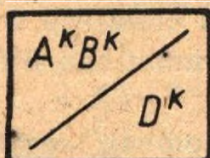
A kivonó egység állapot diagramját két pozitív előjelű binárisan kódolt szám kivonására rajzoljuk fel (4-27. ábra). A kivonó egység is két belső állapottal (q_0 , q_1) rendelkezik annak megfelelően, hogy az a kivonás során a következő, magasabb helyértékről kell-e a kisebbitendőből 1 jegyet átvenni vagy nem (átvétel vagy más szóval áthozat keletkezik-e vagy sem). A két belső állapotot az állapotdiagram felrajzolása során a következőképpen értelmezzük:

Rendszer állapota	Előző helyértékről átvétel
q_0	van
q_1	nincs

A különbség k-adik helyértéken létrejövő D^k számjegye úgy nyerhető, hogy az A^k -ből kivonjuk B^k -t és a (k-1)-edik helyértéken keletkező áthozatot. Az áthozatot kivonás esetén a következő, eggyel magasabb helyértéken vesszük figyelembe.

A két belső állapotot egyetlen tárolóval valósítjuk meg. A szekunder változót képező R tároló két állapotát a két belső állapothoz a következőképpen rendeljük hozzá (állapot kódolás):

Belső állapot	R
q_0	0
q_1	1



KIINDULÓ ÁLLAPOT	$A^k B^k$			
	00	01	10	11
q_0	q_0	q_1	q_0	q_0
q_1	q_1	q_1	q_0	q_1
KÖVETKEZŐ ÁLLAPOT				

$A^k B^k$			
00	01	10	11
0	1	1	0
1	0	0	1
KÜLÖNBSÉG: D^k			

4-27. ábra

Soros kivonó állapot diagramja és állapot táblázata

A kódolt állapot táblázat alapján megszerkesztett igazságtáblázatból az R tároló állapotegyenlete a 4-28. ábra Veitch diagramja alapján:

$$R^k = (\overline{A^k} + B^k) R^{k-1} + \overline{A^k} B^k \overline{R^{k-1}} = (\overline{A^k} + B^k) R^{k-1} + \overline{A^k} B^k$$

A C_1 és C_2 vezérlési függvények:

$$C_1 = \overline{A^k} + B^k \quad \text{és} \quad C_2 = \overline{A^k} B^k$$

Mivel $\overline{C_1}C_2 = 0$, pl. RS típusú tárolót alkalmazva statikus bemeneteinek vezérlése:

$$G_0 = \overline{C_1} = A^k \overline{B^k} \quad \text{és} \quad G_1 = C_2 = \overline{A^k} B^k$$

A különbség:

$$D^k = A^k \oplus B^k \oplus R^{k-1}$$

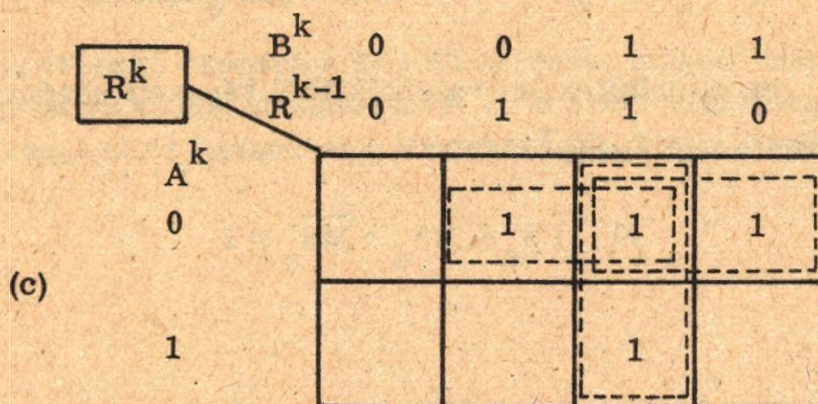
A kivonó egység tervezése a továbbiakban az összeadó egység tervezéséhez hasonlóan végezhető.

(a)

	$A^k B^k$			
R^{k-1}	00	01	10	11
0	0	1	0	0
1	1	1	0	1
KÖV. ÁLLAPOT: R^k				

	$A^k B^k$			
	00	01	10	11
0	0	1	1	0
1	1	0	0	1
KÜLÖNBSÉG: D^k				

(b)	INDEX	A^k	B^k	R^{k-1}	R^k	D^k
	0	0	0	0	0	0
	1	0	0	1	1	1
	2	0	1	0	1	1
	3	0	1	1	1	0
	4	1	0	0	0	1
	5	1	0	1	0	0
	6	1	1	0	0	0
	7	1	1	1	1	1



4-28. ábra

Kivonó (a) kódolt állapot táblázata, (b) igazságtáblázata, (c) Veitch diagramon R^k meghatározásához

4.5.3. Összeadó-kivonó egység

Az A és B binárisan kódolt számot ábrázoljuk előjellel (A_e , B_e) és abszolút értékkel ($|A|$ ill. $|B|$). Az M műveletvezérlő tároló állapotától függően az A és B számot össze kell adni, vagy ki kell vonni egymásból. Legyen a pozitív számok előjelbitje 0, negatív számoké 1. Ha az A és B számmal elvégzendő művelet összeadás, legyen $M = 0$, kivonás esetén 1. Az aritmetikai egység a vezérléstől függően összeadást vagy kivonást végez. Megállapítandó, hogy az A_e , B_e , M értékétől függően mikor kell az aritmetikai egységnek A és B abszolút értékével összeadást, mikor kell kivonást végeznie.

Az összeadás ill. kivonás feltételét az alábbi igazságtáblázatból olvashatjuk ki:

Műveleti utasítás	A_e	B_e	Elvégzendő művelet
összeadás $M = 0$	0	0	$ A + B $
	1	1	
	0	1	$ A - B $
	1	0	
kivonás $M = 1$	0	0	$ A - B $
	1	1	
	0	1	$ A + B $
	1	0	

A 4-29. ábra Veitch diagramjából megállapíthatóan az abszolút értékeket ki kell vonni egymásból, ha

$$M \oplus A_e \oplus B_e = 1 \quad \text{ill.} \quad M \oplus S_e = 1$$

ahol $S_e = A_e \oplus B_e$ az előjelbitek antivalenciájából képezett segédváltozó.

Az abszolút értékeket össze kell adni, ha:

$$M \odot S_e = MS_e + \overline{M}\overline{S_e} = 1$$

**ABSZOLUT ÉRTÉKEK
ÖSSZEADÁSA**

	A_e	0	0	1	1
	B_e	0	1	1	0
M					
0		1		1	
1			1		1

$$M \oplus A_e \oplus B_e = 1$$

**ABSZOLUT ÉRTÉKEK
KIVONÁSA**

	A_e	0	0	1	1
	B_e	0	1	1	0
M					
0			1		1
1		1		1	

$$\overline{M} (\overline{A_e} B_e + A_e \overline{B_e}) + M (\overline{A_e} \overline{B_e} + A_e B_e) = M \odot S_e = 1$$

4-29. ábra
Összeadás ill. kivonás feltétele

4.5.4. Gyakorló feladatok

4.5.4.1. Aritmetikai egység tervezése

1. Tervezendő csoportos átvitelképzéssel működő teljes összeadó egység. Az összeadó k-adik ütemben az A és B szám számjegyeinek kettős csoportjaival végzi el az összeadást a következő táblázatban látható módon:

A^k	B^k	S^k	C^k
00	00	00	0
	01	01	0
	10	10	0
	11	11	0
01	00	01	0
	01	10	0
	10	11	0
	11	00	1
10	00	10	0
	01	11	0
	10	00	1
	11	01	1
11	00	11	0
	01	00	1
	10	01	1
	11	10	1

Rajzoljuk meg a teljes összeadó állapot diagramját, állapot táblázatát. A teljes összeadó az A és B szám számjegyeinek kettes csoportjait összeadja és az eredményt módosítja az előző, alacsonyabb helyértékről érkező átviteli impulzussal. Határozzuk meg a teljes összeadó igazságtáblázatát.

2. Tervezendő csoportos átvitelképzéssel működő párhuzamos üzemi összeadó egység az előző feladat igazságtáblázatának felhasználásával.

A felhasznált tárolók típusa: RS, T, ill. D.

Határozzuk meg a tárolók vezérlésére szolgáló bemeneti egyenleteket, rajzoljuk fel a soros üzemi összeadó egység logikai hatásvázlatát.

3. Tervezendő csoportos áthozat (átvétel) képzéssel működő teljes kivonó egység. A kivonó a k-adik ütemben az A és D szám számjegyeinek kettős csoportjait vonja ki egymásból.

Rajzoljuk meg a teljes kivonó állapot diagramját és állapot táblázatát (az előző, alacsonyabb csoportból érkező áthozat figyelembevételével). Határozzuk meg a teljes kivonó igazságtáblázatát.

4. Párhuzamos üzemi összeadás meggyorsítására szolgál a rekurzív átvitelképzés. Tervezendő az operandusok 4 bites csoportjára átvitelt előállító logikai hálózat, amely az A^k , B^k és a C^{k-1} ismeretében valamennyi helyértékre (pl. az összeadás megkezdése előtt) előállítja a C^k átviteli impulzust. A legalacsonyabb helyértéken az operandusok számjegye A^1 , B^1 , az átvitel $C^0 = 0$.

A következő helyérték átviteli impulzus A^2 , B^2 , A^1 és B^1 segítségével már előállítható (C^2) stb.

Rajzoljuk fel az operandusok 4 bites csoportjához tartozó átvitelképző hálózat logikai hatásvázlatát.

5. Megtervezendő párhuzamos üzemi összeadó egység i -edik fokozata, amely az A^i , B^i számjegyekhez hozzáadja az előző, alacsonyabb helyértékről érkező C^{i-1} átviteli impulzust:

$$V_+ : (A^i) + (B^i) + (C^{i-1}) \rightarrow A_i$$

majd az összeget az akkumulátor regiszterben tárolja.

Az átviteli impulzust kombinációs hálózattal állítsuk elő.

Határozzuk meg az akkumulátor regiszter bemenetének vezérlésére szolgáló logikai egyenleteket, ha az akkumulátor regiszter

- a) T, illetve inverz T
- b) RS, illetve inverz RS

tipusu tároló.

6. Megtervezendő párhuzamos üzemi, aszinkron működésű összeadó egység i -edik fokozata. Az aritmetikai egység az A^i , B^i számjegyekhez hozzáadja az előző alacsonyabb helyértékről érkező C^{i-1} átviteli impulzust:

$$V_+ : (A^i) + (B^i) + (C^{i-1}) \rightarrow A_i$$

és az összeget az akkumulátor regiszterben tárolja.

Az átviteli impulzust kombinációs hálózattal állítsuk elő.

Miután valamennyi helyértéken az összeadás befejeződött, a párhuzamos üzemi összeadó egység lejelentő impulzust állít elő.

Megtervezendő az átvitel, a lejelentő impulzus előállítására szolgáló kombinációs hálózat. Határozzuk meg az akkumulátor regiszter bemenetének vezérlésére szolgáló logikai egyenleteket, ha az akkumulátor regiszter:

- a) T illetve inverz T
- b) JK illetve inverz JK

tipusu tároló.

7. Megtervezendő párhuzamos üzemi, bináris kódolású kivonó egység i -edik fokozata az 5. feladat követelményeivel, amely az A^i és B^i számjegyek különbségéből kivonja az előző, alacsonyabb helyértéken keletkező R^{i-1} áthozatot:

$$V_- : (A^i) - (B^i) - (R^{i-1}) \rightarrow A^i$$

és az eredményt az akkumulátor regiszterben tárolja.

8. Megtervezendő párhuzamos üzemi, bináris kódolású kivonó egység i -edik fokozata a 6. feladat követelményeivel, amely az A^i és B^i számjegyek különbségéből kivonja az előző alacsonyabb helyértéken keletkező R^{i-1} áthozatot:

$$V_- : (A^i) - (B^i) - (R^{i-1}) \rightarrow A^i$$

majd az eredményt az akkumulátor regiszterben tárolja.

9. Tervezzünk két pozitív, " n " jegyű, bináris kódolású, előjellel és abszolút értékkel adott szám összeadására ill. kivonására alkalmas párhuzamos üzemi aritmetikai egységet. A két operandust egy M műveletvezérlő bille-nő kör állapotától függően össze kell adni, vagy ki kell vonni egymásból:

$$M : (A^i) + (B^i) + (C^{i-1}) \rightarrow A^i$$

$$\overline{M} : (A^i) - (B^i) - (C^{i-1}) \rightarrow A^i$$

A feladat követelményei megegyeznek a 6. ill. 8. feladatnál leírtakkal.

10. Binárisan kódolt számok összeadása során keletkező egyetlen hibát párosságvizsgálat segítségével jelezhetünk. Legyenek az operandusok számjegyei:

$$A \doteq (A^n, A^{n-1}, \dots, A^2, A^1)$$

és

$$B \doteq (B^n, B^{n-1}, \dots, B^2, B^1)$$

Az összeadások során az egyes helyértékeken keletkező átvitelek:

$$C = (C^n, C^{n-1}, \dots, C^2, C^1)$$

Az összeg számjegyeit jelöljük

$$S = (S^n, S^{n-1}, \dots, S^2, S^1)$$

módon. Amennyiben az egyes párosság jelző biteket a következőképpen képezzük:

$$P_A = A^1 \oplus A^2 \oplus \dots \oplus A^n; \quad P_B = B^1 \oplus B^2 \oplus \dots \oplus B^n$$

$$P_C = C^1 \oplus C^2 \oplus \dots \oplus C^n; \quad P_S = S^1 \oplus S^2 \oplus \dots \oplus S^n$$

akkor igazolandó, hogy:

$$P_S = P_A \oplus P_B \oplus P_C$$

11. Tervezendő párhuzamos üzemmódban dolgozó, binárisan kódolt decimális (BCD) számok összeadására alkalmas aritmetikai egység. A két BCD kódban ábrázolt szám számjegyeit bináris összeadó egység adja össze decimális számjegyenként. A bináris összeadó kimenetéhez korrigáló logikai hálózat csatlakozik, mely a számjegyek összegének bináris kódját BCD kóddá alakítja át. Ha az A^i és B^i decimális számjegyek összege 9-nél kisebb, korrekcióra nincs szükség. 9-nél nagyobb összeg esetén a korrekciót az alábbi táblázatból határozhatjuk meg:

Az összeg decimális számjegye	Az összeg bináris kódja	Előállítandó BCD kód	Korrekció
10	1010	1 0000	+0110
11	1011	1 0001	
12	1100	1 0010	
13	1101	1 0011	
14	1110	1 0100	
15	1111	1 0101	
16	1 0000	1 0110	
17	1 0001	1 0111	
18	1 0010	1 1000	

Amennyiben a bináris összeadó kimenetén megjelenő ideiglenes összeg 9-nél nagyobb, az ideiglenes összeghez 0110 korrekciót hozzáadva az összeget BCD kódban kapjuk.

Határozzuk meg a decimális átvitel feltételét.

Tervezzük meg az összeadó egység logikai hatásvázlatát. Ehhez használjuk fel az összeadó azon sajátosságát, hogy korrekcióra csak akkor van szükség, ha a következő decimális helyértékre átviteli impulzus kerül.

Az átviteli impulzust állítsuk elő a korrekció előtt. A felhasználható tárolók T típusúak.

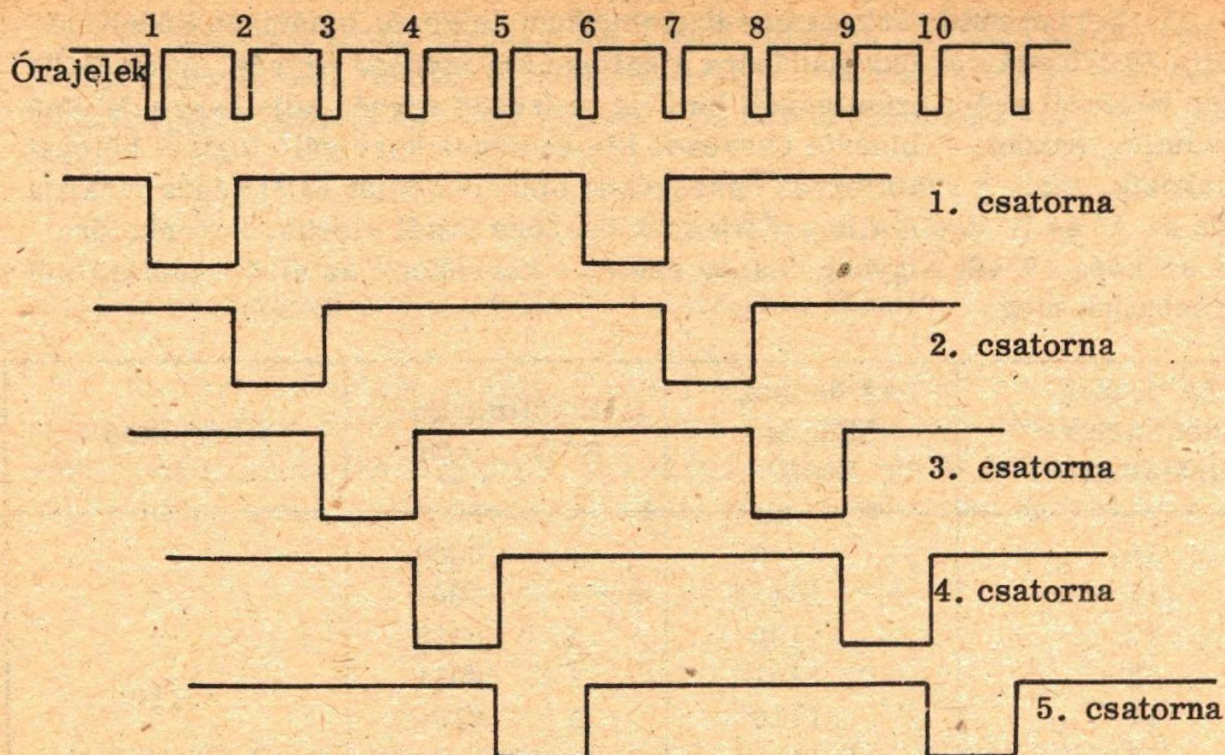
12. Tervezendő párhuzamos üzemmódban dolgozó, binárisan kódolt decimális (BCD) számok kivonására alkalmas aritmetikai egység. A 11. feladatban leírt gondolatmenet felhasználásával oldjuk meg a feladatot.

A felhasználható tárolók T ill. inverz T típusú flip-floppok.

4.5.4.2. Gyakorló feladatok

1. Tervezendő vezérlőjelek előállítására szolgáló egység, amely az órajelek ütemében az 1, 2, 3, 4, 5, 1, ... csatornákra 4-30. ábrán feltüntetett jeleket adja ki.

Határozzuk meg a tárolók bemenetének vezérlésére szolgáló logikai egyenleteket, ha a tárolók RS, T ill. D típusú flip-floppok.



4-30. ábra

2. Tervezendő időpont kiválasztására szolgáló áramkör, amely egy indítójellel meghatározott időponttól kezdve előre beállítható számú órajel beérkezése után vezérlőjelet állít elő. A számlálandó órajel maximális száma: 256.

Az áramkört tervezzük meg

- a) időpontot tároló szelekciós regiszter és komparátor felhasználásával,
- b) a számlánca előre beírt alkalmas kiegészítő szám felhasználásával.

Rajzoljuk meg a feladat idődiagramját. Tervezzük meg a tárolók vezérlésére szolgáló kombinációs hálózatot. A tárolók típusát úgy válasszuk meg, hogy a logikai hálózat a leggazdaságosabb felépítésű legyen.

3. Tervezendő időtartam kiválasztására szolgáló logikai hálózat, amely indítójellel meghatározott időponttól kezdődően előre beállítható N_1 számú órajel beérkezése után V_1 jelet, N_2 számú órajel beérkezése után V_2 vezérlőjelet állít elő. N_1 maximális értéke 512, N_2 maximális értéke 128.

Az áramköri készletet úgy válasszuk meg, hogy a logikai hálózat felépítése a leggazdaságosabb legyen.

Rajzoljuk meg a feladat idődiagramját. Tervezzük meg a tárolók vezérlésére szolgáló kombinációs hálózatot.

4. Megtervezendő egy lyukszalag olvasó vezérlő áramköre az alábbi követelmények alapján:

1. Az olvasó berendezés fedelének zárása után érkező START jel hatására a szalag továbbítás megkezdődik. (Továbbítást vezérlő mágnes meghuz.)
2. Az információ a kimeneten csak akkor jelenhet meg, ha
 - a) START jel után a következő karaktert képviselő lyukak az olvasófejek alá kerültek. Ezt úgy biztosítjuk, hogy a kiolvasás csak a START jel után τ késleltetési idővel kezdődhet.
 - b) A kiolvasott kód legalább egy "1"-est tartalmaz. (Az olvasófej alá lyuk kerül.)
 - c) A továbbító lyuk az olvasófej alatt van.
3. Ha a továbbító lyuk az olvasófejek alól eltávozik, a szalag továbbítást vezérlő mágnes elenged, a fékmágnes meghuz, a szalag leáll.
4. Ujabb START jel hatására a lyukszalag olvasó a következő karaktert olvassa ki.

5. Megtervezendő egy szinkron vezérlésű számítógép művelet dekódoló mátrixa, amely az utasítás műveleti kódjából a makroműveletek elvégzését vezérlő jeleket állítja elő. Az utasítás műveleti kódja 6 bitből áll. A művelet dekódoló kimeneteinek száma: 2^6 .

6. Tervezendő egy kommutáló mátrix, melynek n számú bemenetére órajellel vezérelt bináris számlánc csatlakozik. A kommutáló mátrix 2^n számú kimenetén egymás utáni sorrendben két órajel közötti időnek megfelelő szélességű impulzusok jelennek meg.

7. Tervezendő egy összehasonlító mátrix, mely két regiszter (A, B) tartalmát hasonlítja össze és három kimenetének valamelyikén a következő esetben jelenik meg jel.

- a) Ha A tartalma kisebb, mint B tartalma, akkor az Y_1 kimeneten.
- b) Ha A tartalma nagyobb, mint B tartalma, akkor az Y_2 kimeneten.
- c) Ha A és B regiszter tartalma megegyezik, akkor az Y_3 kimeneten.

8. Tervezzük meg egy kiválasztó mátrix logikai hatásvázlatát. A mátrixnak két bemenete van. Az egyik bemenetére "n" jegyű bináris kódot vezetünk, amely a kiválasztandó szó vagy csatorna címe lehet. Másik bemenetén a bemenő csatornák száma 2^n . A kiválasztó mátrixnak egyetlen kimenő csatornája van. A kimeneten azon csatorna információ tartalma jelenik meg, melynek címét az "n" számú bemenetre kapcsoltuk.

9. Tervezzük meg egy elosztó mátrix logikai hatásvázlatát. A mátrixnak két bemenete van. Az egyik bemenet egyetlen csatornából áll, erre kerül a jelforrás, melynek jelét 2^n számú kimenő csatorna valamelyikére kell kapcsolnunk. Az elosztó mátrix másik bemenete "n" számú csatornából áll, melyre

"n" jegyű bináris kódot vezetünk. Az "n" jegyű bináris kód a kiválasztandó kimenő csatorna címe.

10. Tervezendő egy kódoló mátrix. A kódoló mátrixnak 2^n számú bemenete és n számú kimenete van. A kimeneten a 2^n számú bemenet valamelyikén adott bemenő információ

- a) bináris kódja
- b) Gray kódja
- c) Aiken kódja

jelenjék meg.

5. SZINKRON SEKVENCIÁLIS HÁLÓZATOK

Szinkron szekvenciális hálózatok tervezése során nincs mindig lehetőségünk fizikai megfontolások alapján a belső állapotok (szekunder változók) számának meghatározására. Az általános tervezés során annyi belső állapotot veszünk fel, amennyi a feladat egyértelmű megoldásához szükséges. Az így felvett belső állapotok segítségével elkészítjük az

- a) állapot diagramot
- b) állapot táblázatot
- c) a felvett belső állapotokhoz tartozó állapot táblázatot egyszerűsítjük
- d) a belső állapotokhoz hozzárendeljük a megvalósításukhoz szükséges tárolók állapotát, vagyis a belső állapotokat kódoljuk
- e) kódolt állapot táblázat alapján elkészítjük a tárolók igazságtáblázatát
- f) az igazságtáblázat segítségével meghatározott C_1 , C_2 vezérlési függvények alapján felírjuk a tárolók bemeneti vezérlésének logikai egyenleteit.

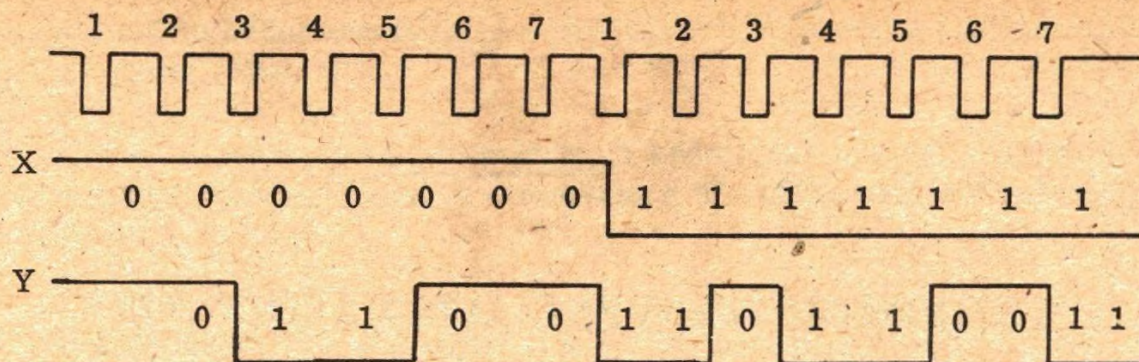
Először a belső állapotok meghatározásával és az állapot táblázat egyszerűsítésével foglalkozunk.

5.1 Szekvenciális hálózatok analízise

Az analízist az 5-1. ábra igazságtáblázatán feltüntetett három tárolóból (Q_1 , Q_2 , Q_3) álló szekvenciális hálózaton végezzük el. A rendszer bemenő jele X, kimenő jele Y. Kérdés: a bemenet-kimenet közötti sorrendi kapcsolatot kevesebb számú belső állapottal nem tudnánk-e megvalósítani?

Az igazságtáblázat (vagy más megadási mód esetén a szekunder változókkal kiegészített ütemdiagram) alapján megrajzoljuk a hálózat állapot táblázatát (5-2. ábra). Az állapot táblázat kódolt belső állapotait az egyszerűbb írásmód kedvéért jelöljük (a, b, c, d, ..., g) betűkkel. Ezzel a jelöléssel a kódolt állapot táblázat alapján meghatározott állapot táblázatot és állapot diagramot az 5-3. ábra mutatja.

A belső állapotok száma az állapot táblázatból megállapíthatóan 7, aminek megvalósításához három szekunder változó (három tároló elem) szükséges.



X^k	k-adik ütem			(k+1)-edik ütem			Y^k
	Q_1^k	Q_2^k	Q_3^k	Q_1^{k+1}	Q_2^{k+1}	Q_3^{k+1}	
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	0	0	1
0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	1	1	1
0	1	1	0	1	0	0	1
0	1	1	1	Tiltott kombináció			
1	0	0	0	0	0	1	0
1	0	0	1	0	1	0	1
1	0	1	0	0	0	1	1
1	0	1	1	1	0	1	0
1	1	0	0	0	0	1	0
1	1	0	1	1	1	0	1
1	1	1	0	1	0	1	1
1	1	1	1	Tiltott kombináció			

5-1. ábra
Szinkron szekvenciális hálózat igazságtáblázata

Belső állapot "k"-adik ütemben $(Q_1 Q_2 Q_3)^k$	Belső állapot a (k+1)-edik ütemben		Kimenet a "k"-adik ütemben	
	$(Q_1 Q_2 Q_3)^{k+1}$		Y^k	
	$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
000	000	001	0	0
001	011	010	1	1
010	100	001	1	1
011	110	101	0	0
100	000	001	0	0
101	011	110	1	1
110	100	101	1	1

5-2. ábra
Szekvenciális hálózat kódolt állapot táblázata

A belső állapotok számának csökkentése az állapot táblázat egyszerűsítésével történhet. Az állapot táblázat egyszerűsítésére két módszert mutatunk be

1. Csoportokra bontást vagy HUFFMAN - MEALY módszert.
2. Lépcsős táblázatok segítségével történő egyszerűsítést.

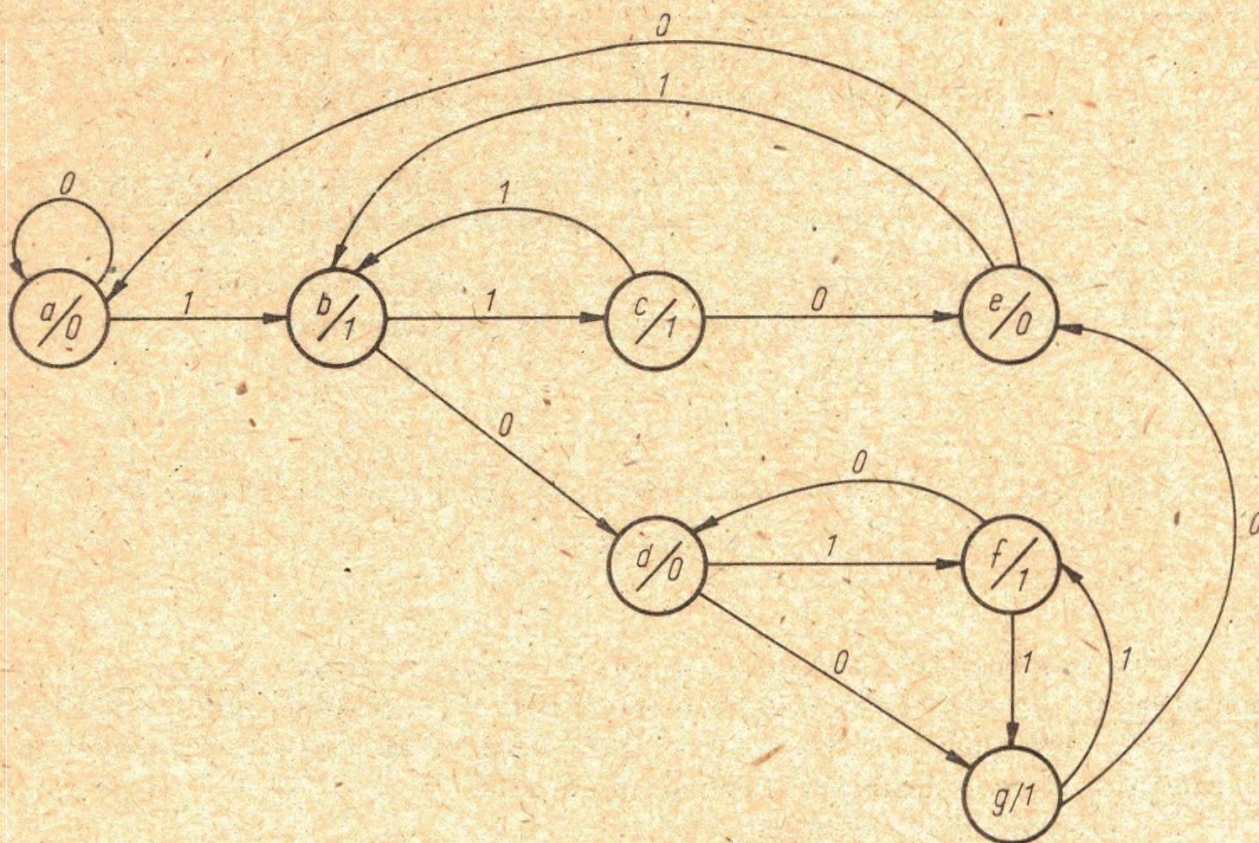
5.2 Állapot táblázat egyszerűsítése HUFFMAN - MEALY módszerrel

Az egyszerűsítés első lépésében megkeressük az állapot táblázat ekvivalens elemeit. Az ekvivalens elemek meghatározását a grafikus minimalizálás (pl. Veitch diagramon történő egyszerűsítés) analógiájára más kifejezéssel az egyeztethető elemek meghatározásának is nevezik.

Két állapot akkor ekvivalens, ha

1. a két állapothoz a bemenő jelek azonos kombinációja tartozik
2. azonos bemenőjel kombinációhoz a kimenőjelek azonos kombinációja tartozik
3. a bemenőjelek azonos kombinációja mellett a (k+1)-edik ütem belső állapotai ekvivalensek.

Belső állapot a "k"-adik ütemben	Belső állapot a (k+1)-adik ütemben		Kimenet a "k"-adik ütemben	
	$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
a	a	b	0	0
b	d	c	1	1
c	e	b	1	1
d	g	f	0	0
e	a	b	0	0
f	d	g	1	1
g	e	f	1	1

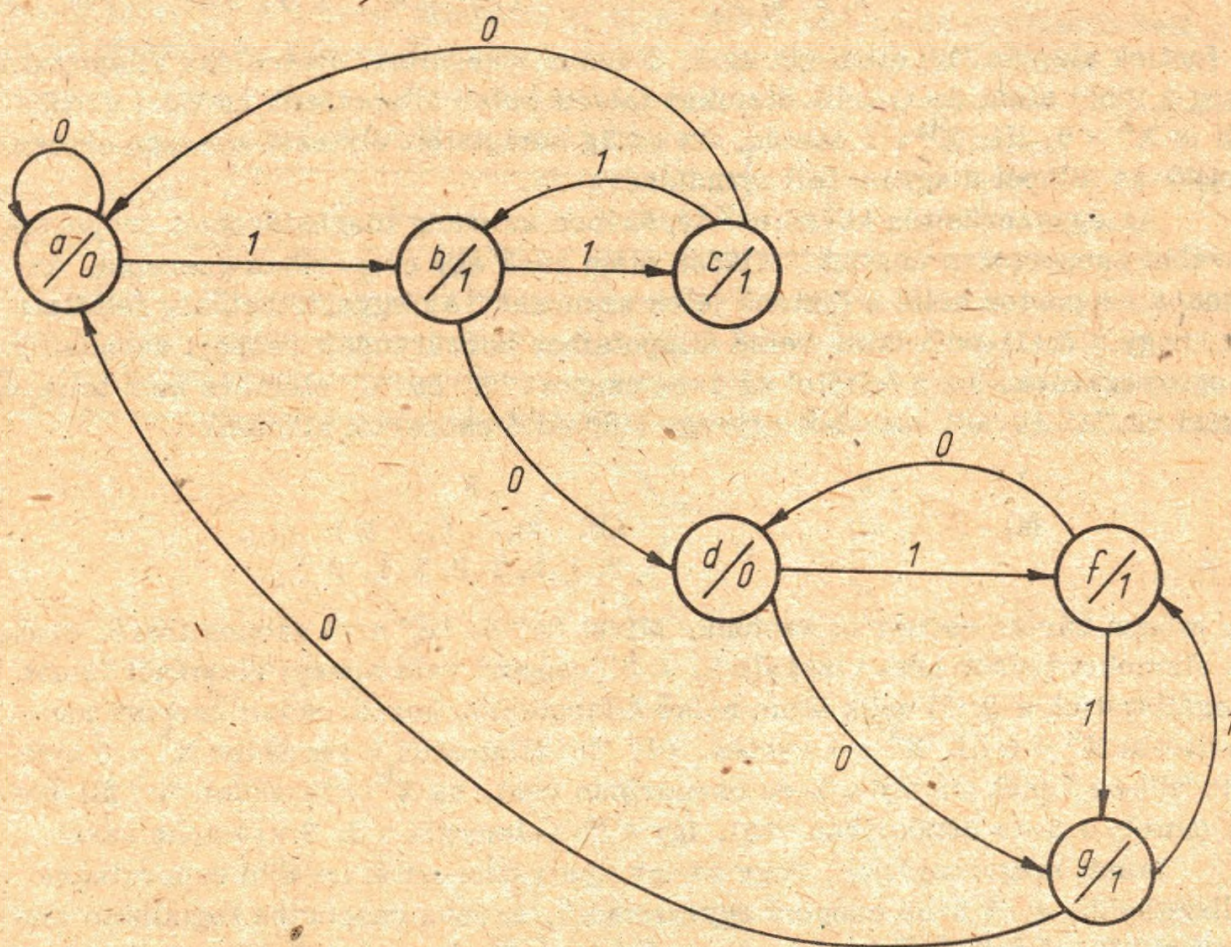


5-3. ábra
Szekvenciális hálózat állapot táblázata és állapot diagramja

Az 5-3. ábra állapot táblázatában az "a" valamint az "e" elemre (belső állapotra) teljesül az ekvivalencia mindhárom feltétele, vagyis: $a \equiv e$. A két ekvivalens állapot közül az egyiket elhagyva az állapot táblázat már csak 6 belső

állapotot tartalmaz. Összetett esetben természetesen az ekvivalens elempárok száma nagyobb lehet. Ebben az esetben minden ekvivalens elempárból egyet megtartva a feltétel nélküli ekvivalens elempárok kiválasztásával egyszerűsítjük tovább az állapot táblázatot. Esetünkben az "e" állapot eliminálásával nyert állapot táblázatot az 5-4. ábra szemlélteti.

$(Q_1 Q_2 Q_3)^k$	$(Q_1 Q_2 Q_3)^{k+1}$		Y^k	
	$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
a	a	b	0	0
b	d	c	1	1
c	a	b	1	1
d	g	f	0	0
f	d	g	1	1
g	a	f	1	1



5-4. ábra
Első lépésben egyszerűsített állapot táblázat és állapot diagram

A csoportokra bontással történő egyszerűsítés első lépésében azonos csoportba foglaljuk azokat az elemeket, melyeknél a kimenő jelek kombinációja azonos. Nyilvánvaló, hogy az így nyert csoportok egyes elemeinél a $(k+1)$ -edik ütem belső állapotai már nem azonosak. Példánkban a 6 elemet így módon két csoportra oszthatjuk a következő jelöléssel:

(a, d)

(b, c, f, g)

Az egyes csoportokat valamilyen (tetszőleges) számmal jelöljük meg a következőképpen:

1

(a, d)

2

(b, c, f, g)

A csoport egyes elemeinél külön feltüntetjük, hogy a bemenőjel hatására létrejövő belső állapotváltozás következtében a vizsgált elem melyik csoportba kerül. Pl. ha $X^k = 0$, akkor "b" állapotból a rendszer "d"-be kerül (amely az 1-es csoportban van); $X^k = 1$ esetén pedig "b"-ből "c"-be (amely a 2-es csoportban van). A $(k+1)$ -edik ütem két lehetséges belső állapotának csoport azonosítóját az elem alatt tüntetjük fel:

1

(a, d)

1, 2 2, 2

2

(b, c, f, g)

1, 2 1, 2 1, 2 1, 2

A fentiek alapján "b" elem alá az 1, 2 számok kerültek. Hasonlóan állapíthatók meg a többi elem $(k+1)$ -edik ütemben felvett belső állapotának csoport azonosítói is $X^k = 0$, ill. $X^k = 1$ esetén. Az eddig bemutatott eljárást könnyen elvégeztethetjük az állapot diagram felhasználásával is.

Az egyszerűsítés következő lépésében az eddig meghatározott csoportokat további csoportokra osztjuk. Minden elem csak egy csoportban szerepel. Az azonos csoporton belül a kimenő jelek azonosak. Az egyeztetetőség feltétele az, hogy a $(k+1)$ -edik ütem belső állapotaihoz hozzárendelt csoport azonosítók megegyezzenek. Ez a feltétel az 1-es csoport "a" és "d" elemére nem teljesül, ezért az "a" és "d" elemből egy-egy különálló csoportot alkotunk:

1

(a)

2

(d)

3

(b, c, f, g)

2, 3 1, 3 2, 3 1, 3

Ily módon három csoportot kaptunk. Mivel "a" és "d" nem azonosítható, ezeket az elemeket tovább nem vizsgáljuk. A 3. csoport valamennyi eleménél ismét megállapítjuk a $(k+1)$ -edik ütem belső állapotaihoz hozzárendelt csoport azonosítókat $X^k = 0$ ill. $X^k = 1$ esetén. Pl. "b" állapotból a rendszer $X^k = 0$ mellett "d"-be kerül (amely a 2-es csoportban van), ha $X^k = 1$, akkor "c" állapotba (amely a 3-as csoportban van). Így a "b" elem alá a 2, 3 számpár kerül.

Ismét megvizsgáljuk, hogy az így nyert csoportok további csoportokra felbonthatók-e. A 3-as csoport egyeztethető, azonos csoportba foglalható elemeit abból a feltételből kapjuk, hogy a $(k+1)$ -edik ütem belső állapotaihoz hoz-

zárendelt azonosítók egyezzenek meg. Ez láthatóan a "b" és "f" ill. "c" és "g" elempárra teljesül. Ezeket külön csoportba foglalva:

1	2	3	4
(a)	(d)	(b, f)	(c, g)
		2, 4 2, 4	1, 3 1, 3

ismét meghatározzuk a $(k+1)$ -edik ütem belső állapotaihoz hozzárendelt csoport azonosítókat.

Az így kapott 4 csoport elemei már egyeztethetők, mert a csoport egyes elemeihez:

- a) a kimenő jelek azonos kombinációja tartozik,
- b) a kimenő jelek kombinációja megegyezik. A csoportokra beosztásnak ez volt az első feltétele.

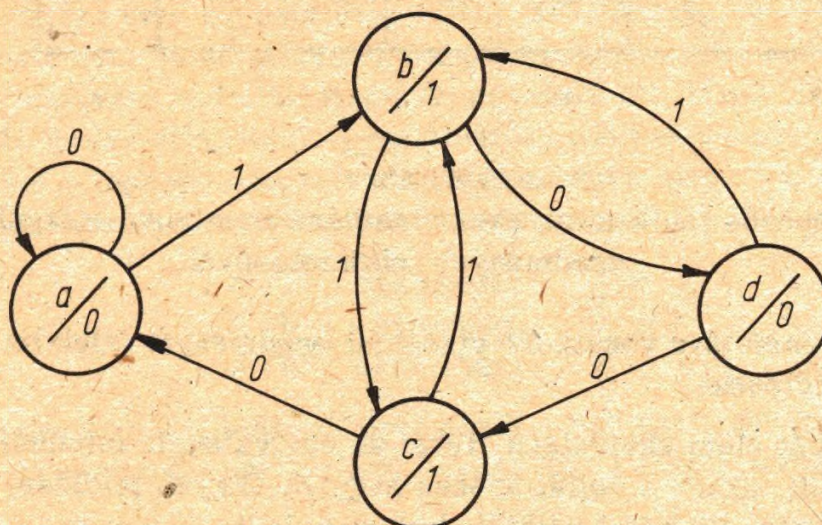
Láthatóan: $b \equiv f$, $c \equiv g$

c) tehát a $(k+1)$ -edik ütem belső állapotai is ekvivalensek.

Az állapot táblázat egyszerűsítése összetettebb feladatok esetén is a bemutatott eljárás alkalmazásával végezhető el.

Az egyszerűsítés eredményeként a rendszer realizálható 4 belső állapottal (7 helyett). A belső állapotok megvalósításához 2 tároló elemre van szükségünk (3 helyett). A 4 belső állapotot a, b, c, d-vel jelölve az egyszerűsített állapot táblázat és állapot diagram könnyen megrajzolható (5-5. ábra).

k-adik ütem belső állapota	(k+1)-edik ütem belső állapota		Y^k kimenet	
	$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
a	a	b	0	0
b	d	c	1	1
c	a	b	1	1
d	c	b	0	0



5-5. ábra

Egyszerűsített állapot táblázat, ill. állapot diagram

5.3 Állapot táblázat egyszerűsítése lépcsős táblázattal

Lépcsős táblázattal történő egyszerűsítés során nincs szükségünk a feltétel nélkül ekvivalens elempárok előzetes megkeresésére, tehát 5-3. ábra állapot táblázatából indulunk ki. (Az egyszerűbb írásmód kedvéért célszerű ebben az esetben is a kódolt belső állapotokat a, b, ..., g betűkkel jelölni.)

A lépcsős táblázat vízszintes és függőleges oszlopán a rendszer k-adik ütemben felvett belső állapotait helyezzük el (5-6. ábra). A vízszintes és függőleges oszlopok kereszteződése egy-egy cellát hoz létre. A cellába a két belső

a							
b	x						
c	x	de					
d	ag bf	x	x				
e		x	x	ag bf			
f	x	cg	de bg	x	x		
g	x	de cf	bf	x	x	de	
	a	b	c	d	e	f	g

5-6. ábra

Lépcsős táblázat az ekvivalenciára vonatkozó vizsgálat
eredmények feltüntetésével

állapot ekvivalenciájára vonatkozó vizsgálat eredményét írjuk be. Szokásosak a következő jelölések:

1. Ha a két elem ekvivalens, tehát az ekvivalencia mindhárom feltétele teljesül, akkor a cellát üresen hagyjuk. Egyéb jelölése: $\odot$, $\equiv$, $\checkmark$ (vonás). (Feltétel nélküli ekvivalencia.)
2. Ha a két elem ekvivalenciájának első két feltétele teljesül, de a (k+1)-

edik ütem belső állapotai már nem azonosak, akkor a cellába beírjuk az egyeztethetőség (ekvivalencia) feltételét. (Feltételes ekvivalencia.)

3. Ha a két elem ekvivalenciájának első két feltétele sem teljesül, vagyis a kimenőjelek kombinációja nem azonos, akkor a két elem nem egyeztethető, ezt a cellába irt X-szel jelöljük.

Az 5-6. ábrán felrajzolt lépcsős táblázatból megállapítható, hogy a függőleges oszlopon az első, a vízszintes oszlopon az utolsó elem elhagyható (vizsgálatokhoz elegendő a vastagon kihúzott egyenesekkel határolt terület). Az ábrán az egyetlen ekvivalens elempárhoz tartozó cellát üresen hagytuk.

Ezek után eldöntendő, hogy a cellákba beírt ekvivalencia feltételek közül melyek teljesülnek, melyek az egyeztethető feltételes ekvivalenciával rendelkező elempárok. A kiinduláshoz a feltételt az antivalens (X-szel jelölt) elempárok adják. Pl. a lépcsős táblából látható, hogy a $\neq g$. Az $a \equiv g$ feltétel szerepel (a, d) és a (d, e) elempárban, tehát mivel: $a \neq g$, ezért

$$a \neq d \quad \text{és} \quad d \neq e$$

Mivel az $(a \neq d)$ követelmény egyetlen feltételes ekvivalens cellában sem szerepel, $a \neq d$ nem eredményez újabb antivalenciákat. De a $(d \neq e)$ feltétel maga után vonja, hogy:

$$f \neq g, \quad c \neq f, \quad b \neq c, \quad b \neq g$$

A vizsgálat eredményét a lépcsős táblázaton úgy tüntetjük fel, hogy a kiinduláskor választott $(a \neq g)$ antivalens cellába még egy X-et helyezünk el (ezzel rögzítettük, hogy ennek az antivalenciának következményeit megvizsgáltuk). Hasonlóképpen áthúzzuk azokat a feltételeket is, ahol $(a \equiv g)$ szerepel. Annak jelölésére, hogy $a \neq d$ további következményeit is megvizsgáltuk, a cellába még egy X-et írunk. A $(d \neq e)$ feltétel vizsgálatát a (d, e) elempárhoz tartozó cellába irt X-szel jelölve, áthúzzuk azokat a feltételeket, ahol $(d \equiv e)$ szerepelt. Az eljárást az elmondottak alapján mindaddig folytatjuk, amíg még nem kerestük az összes össze nem egyeztethető cellát (5-7. ábra).

Az egyeztethető belső állapotok meghatározása ezek után a feltételes és feltétel nélkül ekvivalens cellák meghatározásából áll. Azok a feltételes ekvivalens cellák (belső állapotok) egyeztethetők, melyekben nincs áthúzás és kereszt. Ezeket a lépcsős diagram alapján keressük meg. Az "f", "e", "d" oszlopban nincsenek összevonható belső állapotok.

$$\begin{array}{ll} \text{A "c" oszlopban:} & c \equiv g \\ \text{A "b" oszlopban:} & b \equiv f \\ \text{Az "a" oszlopban:} & a \equiv e \end{array}$$

Mivel a 7 kiindulási belső állapot közül a "d" egyik feltételben sem szerepel, az egyik elemmel sem vonható össze, "d"-t még külön szerepeltetnünk kell.

Tehát az egyszerűsítések eredményeképpen négy belső állapottal realizálható a rendszer, ezek: a, b, c, d.

b	xx					
c	xx	de x				
d	de x	xx	xx			
e		xx	xx	de x		
f	xx	cg	de x	xx	xx	
g	xx	de x	bf	xx	xx	de x
	a	b	c	d	e	f

5-7. ábra
Antivalens cellák meghatározása

$(AB)^k$	$(AB)^{k+1}$		Y^k	
	$X^{k=0}$	$X^{k=1}$	$X^{k=0}$	$X^{k=1}$
00	00	01	0	0
01	11	10	1	1
10	00	01	1	1
11	10	01	0	0

index	k-adik ütem			(k+1)-edik ütem		k-adik ütem
	X^k	A^k	B^k	A^{k+1}	B^{k+1}	Y^k
0	0	0	0	0	0	0
1	0	0	1	1	1	1
2	0	1	0	0	0	1
3	0	1	1	1	0	0
4	1	0	0	0	1	0
5	1	0	1	1	0	1
6	1	1	0	0	1	1
7	1	1	1	0	1	0

5-8. ábra
Egyszerűsített kódolt állapot táblázat és igazság táblázat

Az 5-5. ábra egyszerűsített állapot táblázatának négy belső állapotát két tároló (A, B) segítségével valósíthatjuk meg. Az állapot kódolás legyen a következő:

Belső állapot	A	B
a	0	0
b	0	1
c	1	0
d	1	1

Az egyszerűsített kódolt állapot tábla alapján meghatározzuk az egyszerűsített igazságtáblázatot (5-8. ábra). Az igazságtáblázatból minimalizálás után:

$$A^{k+1} = A^k B^k \overline{X^k} + \overline{A^k} B^k$$

$$B^{k+1} = (A^k \overline{X^k} + \overline{A^k} \overline{X^k}) B^k + \overline{B^k} X^k$$

Tehát a tárolók vezérlési függvényei:

$$C_{1A} = B^k \overline{X^k} \quad \text{és} \quad C_{2A} = B^k$$

illetve

$$C_{1B} = A^k \odot X^k \quad \text{és} \quad C_{2B} = X^k$$

A négy belső állapotot pl. JK típusú tárolóval realizálva a két tároló bemenetének vezérlésére szolgáló logikai egyenletek:

$$K_A = \overline{C_{1A}} = \overline{B^k} + X^k \quad \text{és} \quad J_A = C_{2A} = B^k$$

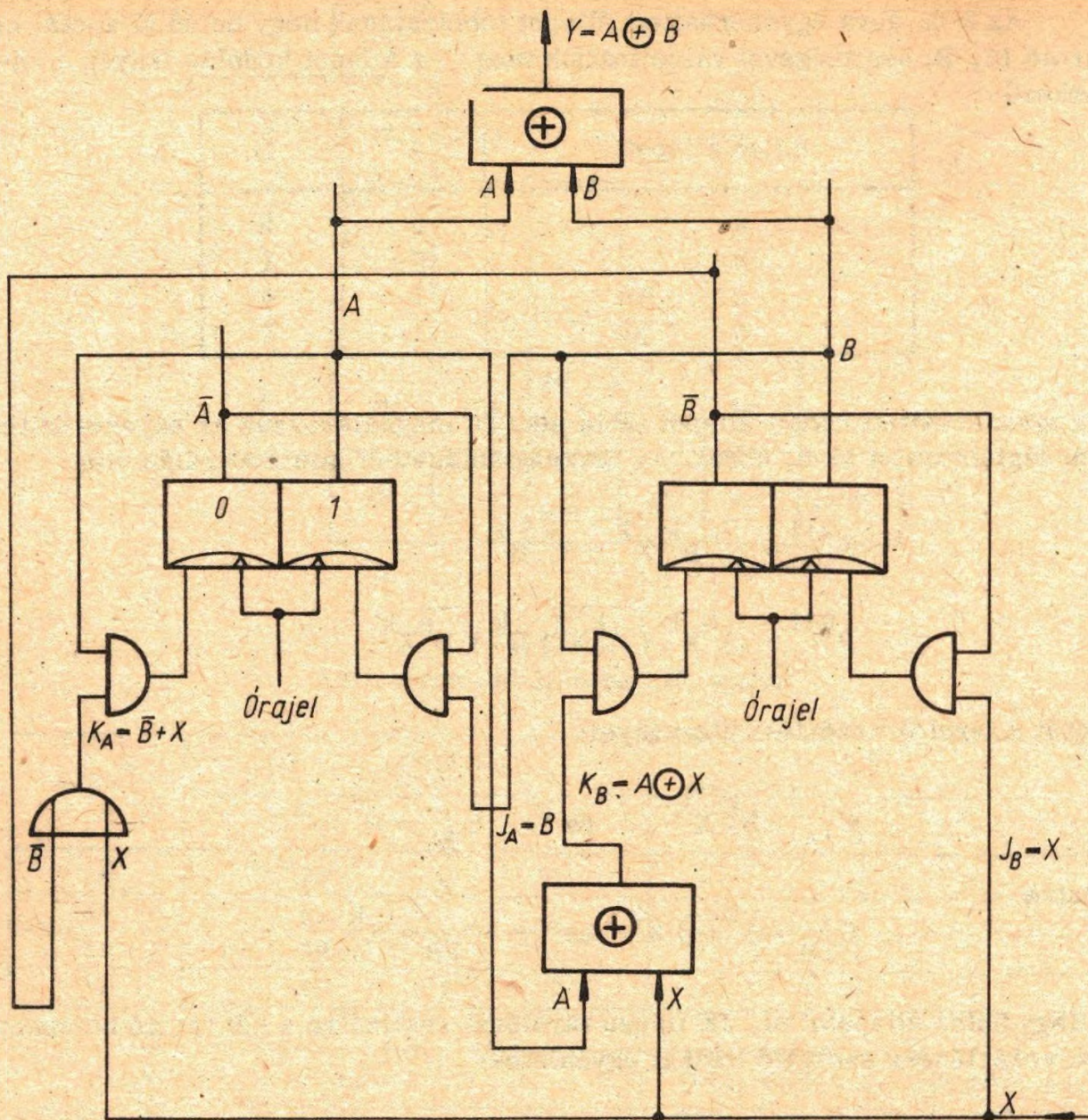
ill.

$$K_B = \overline{C_{1B}} = A^k \oplus X^k \quad \text{és} \quad J_B = C_{2B} = X^k$$

A kimenő változó logikai egyenlete minimalizálás után:

$$Y^k = A^k \oplus B^k$$

Az analizált szekvenciális hálózat elvi logikai hatásvázlatát az 5-9. ábra mutatja.



5-9. ábra

Az analizált szekvenciális hálózat elvi logikai hatásvázlata

5.4 Szekvenciális hálózatok tervezése

5.4.1. Egyszerű hibajelző

a) A feladat megfogalmazása

Tervezzünk egy soros működésű hibajelző áramkört, melynek kimenetén 1101, 1110, 1111 tilos bemenő jel kombinációk esetén jel jelenik meg és egyidejűleg az áramkör alaphelyzetbe tér vissza. Alaphelyzetben (az első beérkező

soros impulzus előtt) a szekvenciális hálózat regiszterének valamennyi tárolója 0-ás állapotban van.

A hibajelző áramkör a három többletes (excess three), más elnevezéssel Stibitz kód érvénytelen kombinációinak jelzésére alkalmas. A három tilos kombináció a $(10 + 3)$, $(11 + 3)$ ill. $(12 + 2)$ decimális számoknak felel meg.

A szekvenciális hálózat a negyedik beérkező impulzus után, az előző három bit ismeretében állapítja meg, hogy tiltott 4 bites kombináció érkezett-e vagy sem. A soros kódban érkező impulzusokat előzetesen tároljuk az érkezés sorrendjében. Amennyiben az előző három impulzus 101 volt, a negyedik pedig 1, akkor 1101 4 bites kombinációt kell vizsgálni. Mivel ez feltételünk szerint tiltott, a szekvenciális hálózat 000 alaphelyzetbe kerül, és a kimeneten jelzést ad. Pl. 011 tárolt érték esetén, 1-es bejövő impulzus hatására az 1011 megengedett kombináció lép fel. A kimeneten hibajel nem keletkezik, de tárolandó az utolsó három impulzus: 101.

b) Belső állapotok felvitele, állapot diagram meghatározása

A leírt működés alapján kíséreljük meg a feladatot megoldani három tároló (8 belső állapot) felvételével. (Elvben lehetőségünk lenne a négy impulzus tárolására is.) A három tárolónak nyolc lehetséges kombinációját jelölhetnénk a tárolók kódolt állapotaival, de az előző fejezetekben láttuk, hogy kényelmesebb az állapotokhoz egy-egy betűt rendelni (5-10. ábra). Az állapot diagram és az állapot táblázat az elmondottak alapján könnyen felrajzolható.

c) Felvett állapot táblázat egyszerűsítése

A tervezés következő lépése a felvett állapot táblázat egyszerűsítése. Az egyszerűsítést végezzük el a lépcsős tábla segítségével (5-11. ábra).

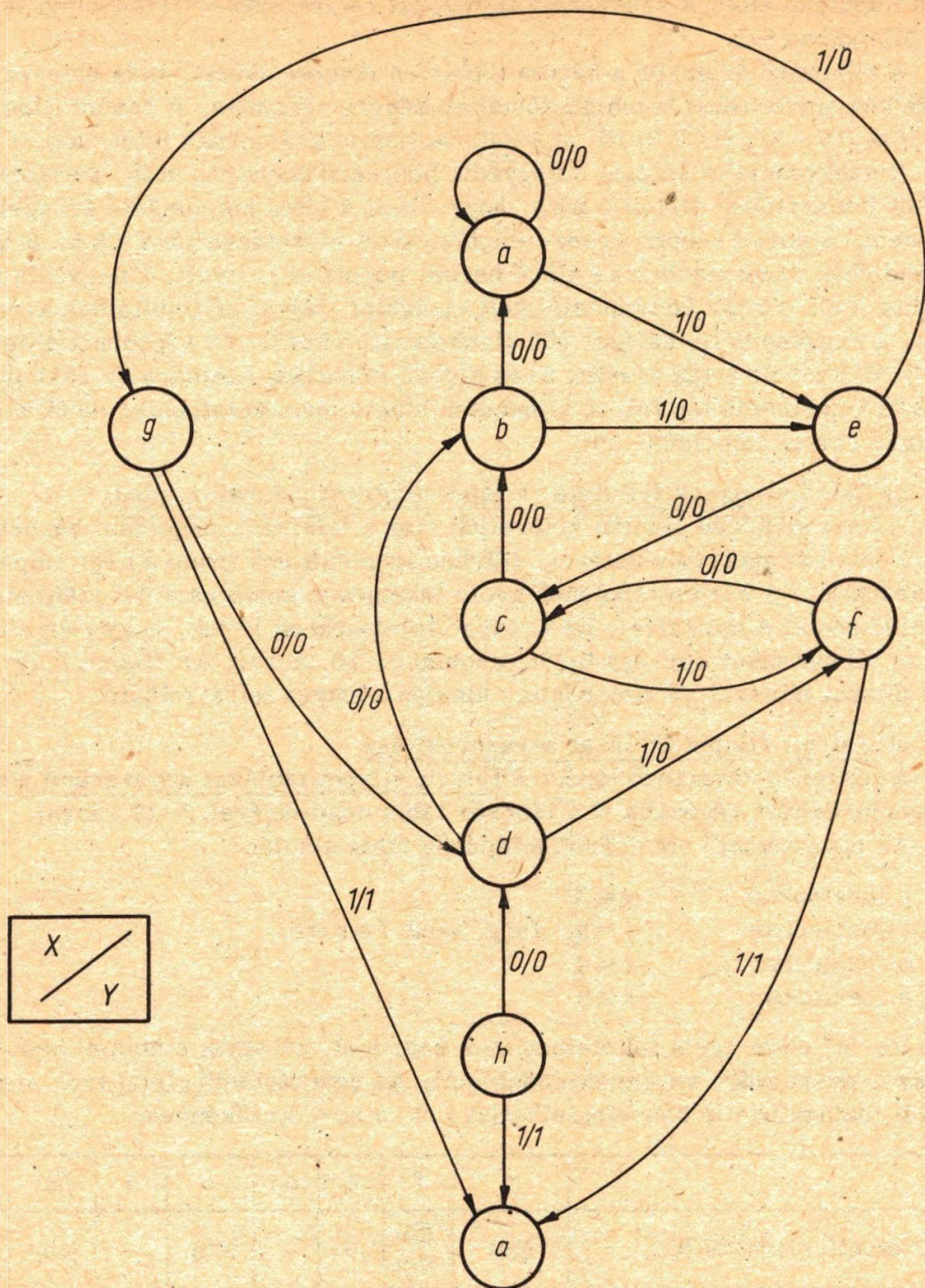
Az egyeztethető elempárok a lépcsős tábla alapján:

g oszlopban	$g \equiv h$
f oszlopban	$f \equiv h, f \equiv g, \text{ tehát } f \equiv g \equiv h$
c oszlopban	$c \equiv d$
a oszlopban	$a \equiv b$

Mivel az "e" elem egyik feltételben sem szerepel, ez egyik elemmel sem vonható össze, ezért külön szerepeltetnünk kell. Az egyeztethető elempárok megkeresését táblázatban is összefoglalhatjuk, pl. a következőképpen:

	g	f	e	d	c	b	a	
Egyeztethető elempárok	(gh)	(fh) (fg)	-	-	(cd)	-	(ab)	
Egyeztethető csoportok	(fgh)				(cd)		(ab)	d
Egyeztethető csoportok jelölése	f				c		a	e

Az egyszerűsített állapot táblázat (4-12. ábra) tehát az (a, c, e, f) elemeket tartalmazza. A kiinduláskor felvett nyolc belső állapot helyett a hálózat négy belső állapottal is realizálható, amit két tárolóval (A, B) realizálhatunk.



5-10/a ábra
Hibajelző áramkör állapot diagramja

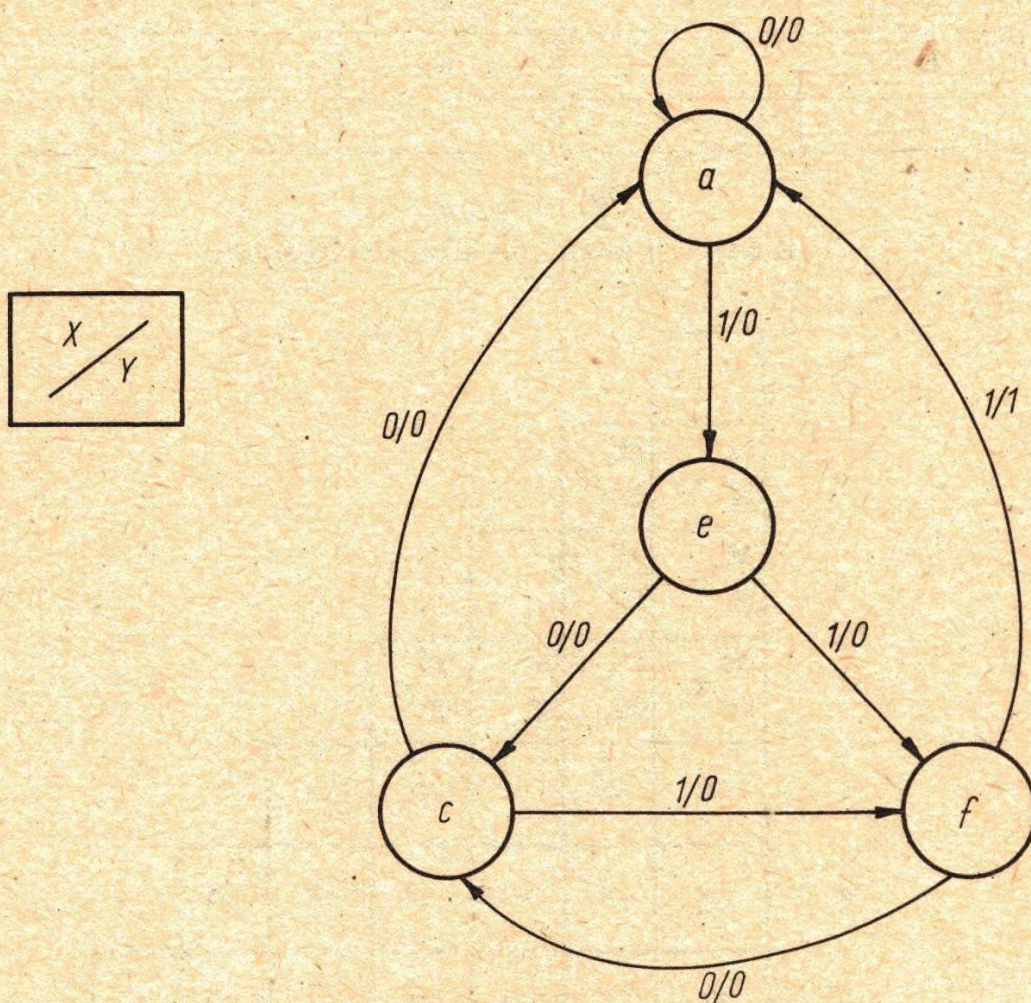
k-adik ütem belső állapota	(k+1)-edik ütem belső állapota		Y^k kimenet	
	$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
a (000)	a	e	0	0
b (001)	a	e	0	0
c (010)	b	f	0	0
d (011)	b	f	0	0
e (100)	c	g	0	0
f (101)	c	a	0	1
g (110)	d	a	0	1
h (111)	d	a	0	1

5-10/b ábra
Hibajelző áramkör állapot táblázata

b							
c	ac x	bc x					
d	ad x	bd x					
e	ae x	be x	ce x	de x			
f	xx	xx	xx	xx	xx		
g	xx	xx	xx	xx	xx	cd	
h	xx	xx	xx	xx	xx	cd	
	a	b	c	d	e	f	g

5-11. ábra
Lépcsős tábla a felvett állapot táblázat egyszerűsítésére

$(AB)^k$	$(AB)^{k+1}$		Y^k	
	$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
a	a	e	0	0
c	a	f	0	0
e	c	f	0	0
f	c	a	0	1



5-12. ábra
Egyszerűsített állapot táblázat, állapot diagram

d) Állapot kódolás

Tervezés következő lépése a belső állapotok kódolása, az állapot kódolás. Ezt végezzük el a következőképpen:

Belső állapot	A	B
a	0	0
c	0	1
e	1	0
f	1	1

e) Realizálás

A kódolt állapot táblázat segítségével meghatározott igazságtáblázat az 5-13. ábrán látható. Az igazságtáblázat alapján:

$$A^{k+1} = A^k \overline{B^k} X^k + A^k X^k$$

$$B^{k+1} = B^k (A^k \oplus X^k) + A^k \overline{B^k}$$

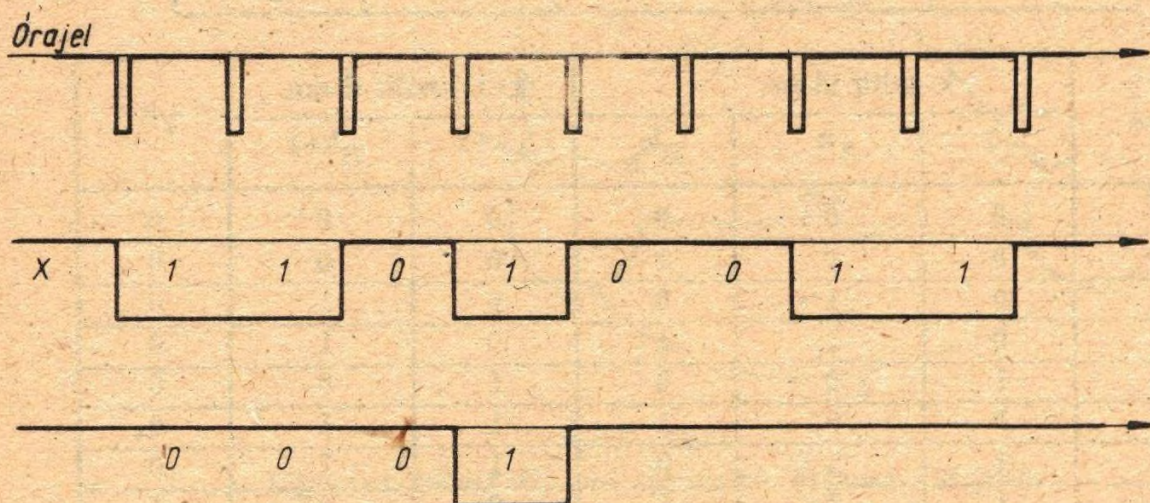
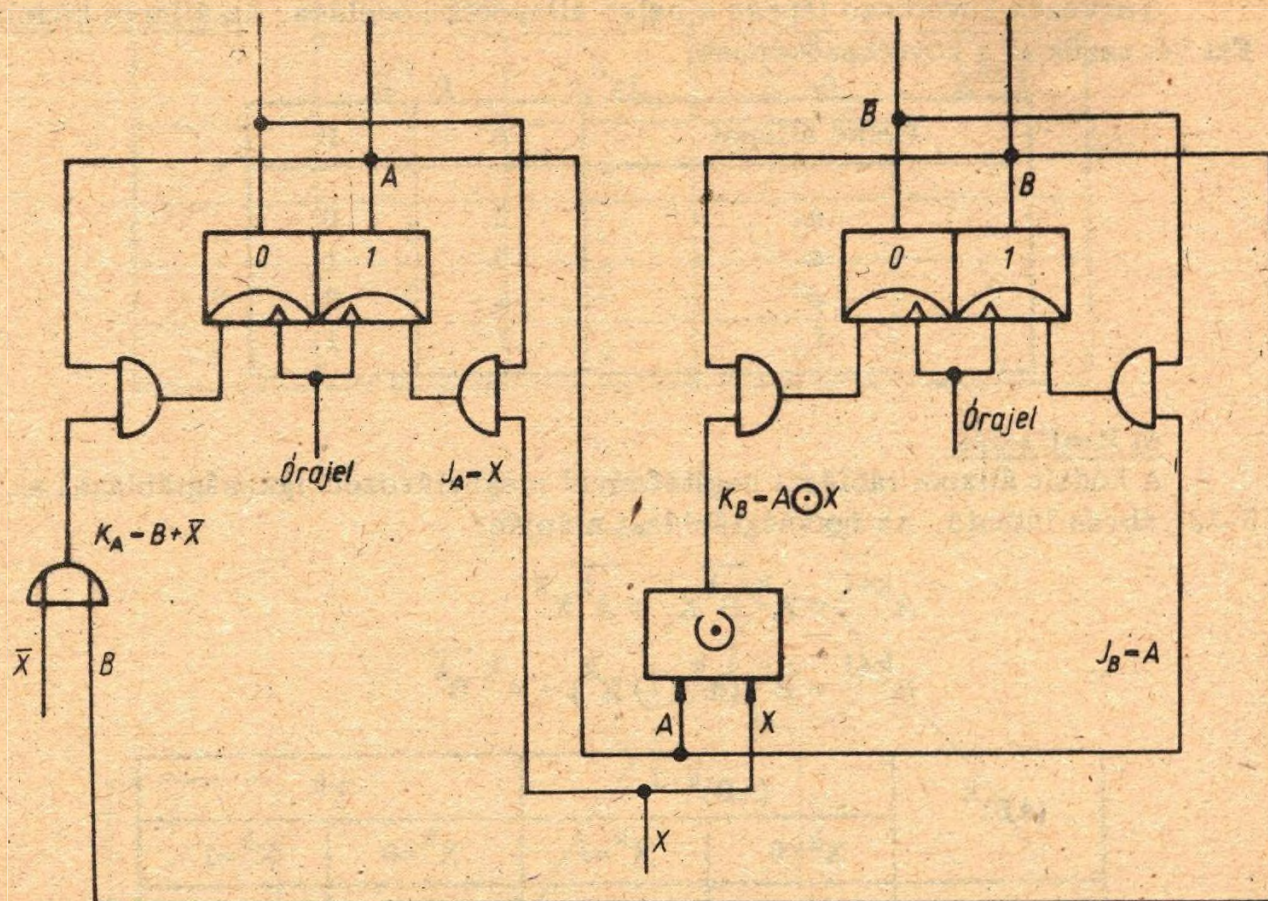
(a)

$(AB)^k$	$(AB)^{k+1}$		Y^k	
	$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
00	00	10	0	0
01	00	11	0	0
10	01	11	0	0
11	01	00	0	1

index	k-adik ütem			(k+1)-edik ütem		Y^k
	X^k	A^k	B^k	A^{k+1}	B^{k+1}	
0	0	0	0	0	0	0
1	0	0	1	0	0	0
2	0	1	0	0	1	0
3	0	1	1	0	1	0
4	1	0	0	1	0	0
5	1	0	1	1	1	0
6	1	1	0	1	1	0
7	1	1	1	0	0	1

5-13. ábra

(a) Egyszerűsített kódolt állapot táblázat,
(b) igazságtáblázat



5-14. ábra
Soros működésű hibajelző szekvenciális hálózat elvi logikai hatásvázlata

A tárolók vezérlési függvényei:

$$C_{1A} = \overline{B^k} X^k \quad \text{és} \quad C_{2A} = X^k$$

illetve

$$C_{1B} = A^k \oplus X^k \quad \text{és} \quad C_{2B} = A^k$$

A négy belső állapotot pl. JK tárolóval realizálva a két tároló bemenetének vezérlését végző logikai egyenletek:

$$K_A = \overline{C_{1A}} = B^k + \overline{X^k} \quad \text{és} \quad J_A = C_{2A} = X^k$$

illetve

$$K_B = C_{1B} = A^k \odot X^k \quad \text{és} \quad J_B = C_{2B} = A^k$$

A kimenő változó logikai egyenlete:

$$Y^k = A^k B^k X^k$$

A megtervezett szekvenciális hálózat elvi logikai hatásvázlata az 5-14. ábrán látható.

5.4.2. Ciklikus hibajelző

Egészítsük ki a hibajelző áramkört újabb három tilos kombinációval, melyek: 0000, 0001, 0010 (az 1, 2, 3 decimális számok binárisan kódolt megfelelői). Feladatunk: ha a következő hat tilos kombináció: 0000, 0001, 0010, 1101, 1110, 1111 valamelyikét észleli a logikai hálózat, akkor a kimeneten jel jelenik meg. A negyedik impulzus beérkezése után a hálózat minden esetben alaphelyzetbe tér vissza! Alaphelyzetben (az első beérkező soros impulzus előtt) a szekvenciális hálózat regiszterének valamennyi tárolója 0-ás állapotban van.

a) Belső állapotok felvétele

A feladat megoldásához vegyünk fel négy tárolót. A négy tároló 16 lehetséges kombinációját jelöljük q_i -vel ($i = 1, \dots, 16$); kiindulási állapotban (q_1) minden tároló 0-ás állapotban van. Az első bejövő impulzust tárolva a regiszter q_2 vagy q_3 állapotba kerül, attól függően, hogy $X^k = 0$ vagy 1.

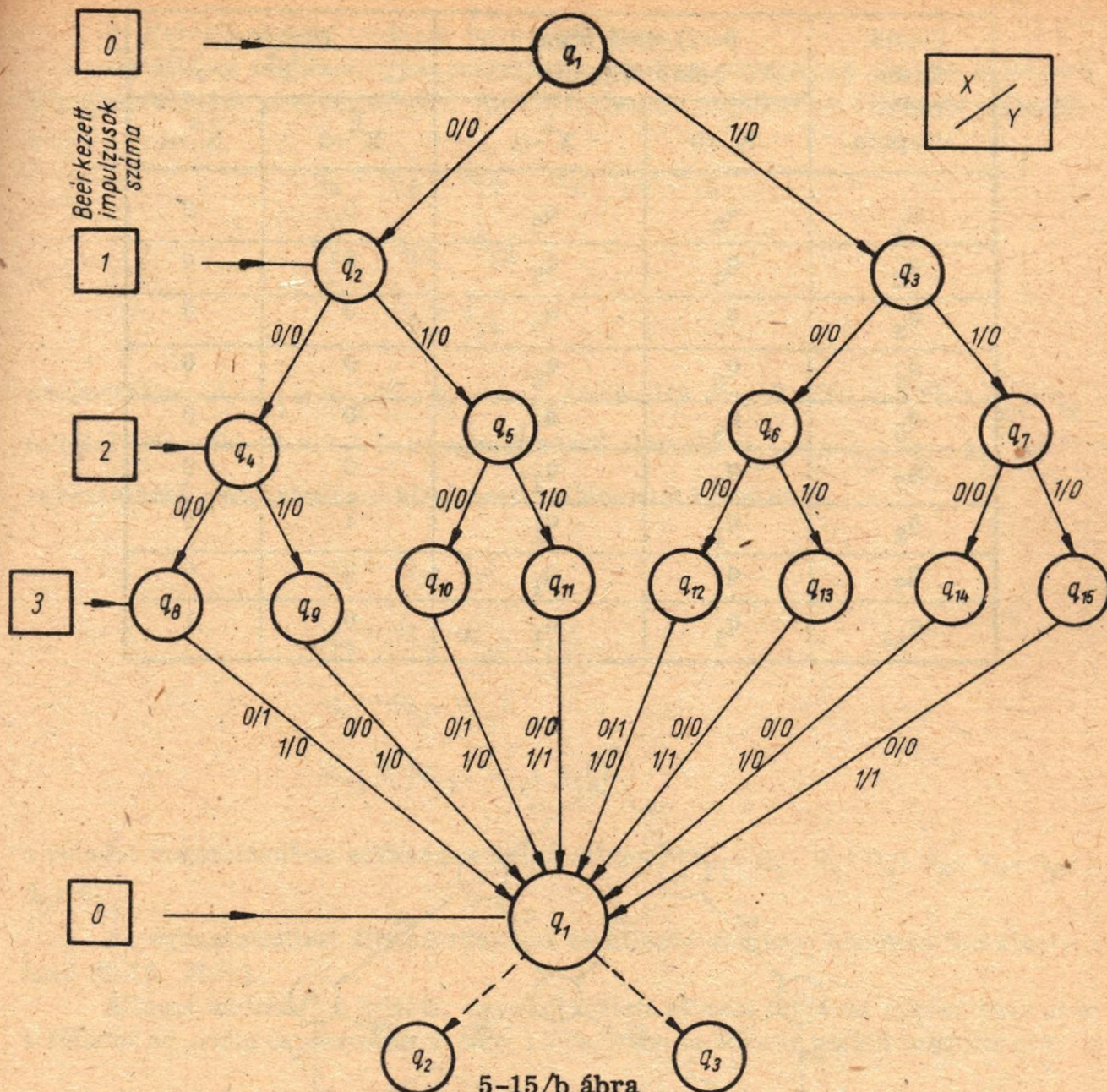
A regiszter első három cellája még üres. A $q_2 = \dots 0$ állapotból a rendszer $X^k = 0$ esetén a $q_4 = \dots 00$, $X^k = 1$ esetén $q_5 = \dots 10$ állapotba, $q_3 = \dots 1$ esetén pedig $q_6 = \dots 01$ vagy $q_7 = \dots 11$ állapotba kerül. Pl. q_6 -ból $X = 0$ esetén

$q_{12} = -001$ ill. $X = 1$ esetén $q_{13} = -101$ állapotba jut. Amennyiben q_{12} állapotban levő rendszer következő soros bemenő jele 0, akkor a négy bitből álló bemenő szó 0001 tiltott kombináció, $Y = 1$ és a rendszer q_1 állapotba tér vissza. $X = 1$ beérkező jel esetén a bemenő kódszó 1001 megengedett kombináció, $Y = 0$, a rendszer most is alapállapotba kerül. A fentiek alapján a hálózat állapot táblázata már elkészíthető (5-15. ábra).

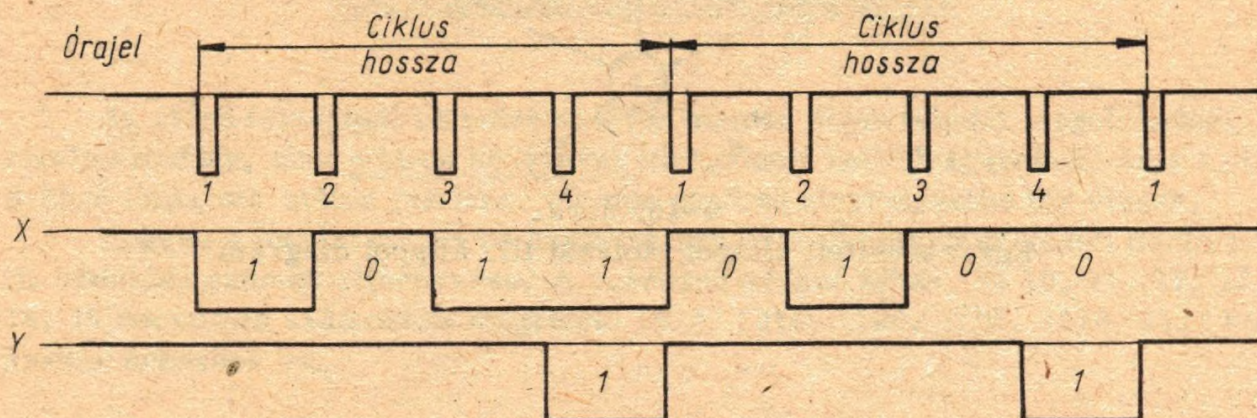
Beérke- zett im- pulzusok száma	k-adik ütem belső állapota	Regiszter állapota	(k+1)-edik ütem belső állapota		Kimenet Y^k	
			$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
0	q_1	----	q_2	q_3	0	0
1	q_2	---0	q_4	q_5	0	0
	q_3	---1	q_6	q_7		
2	q_4	--00	q_8	q_9	0	0
	q_5	--10	q_{10}	q_{11}		
	q_6	--01	q_{12}	q_{13}		
	q_7	--11	q_{14}	q_{15}		
3	q_8	-000	q_1	q_1	1	0
	q_9	-100			0	0
	q_{10}	-010			1	0
	q_{11}	-110			0	1
	q_{12}	-001			1	0
	q_{13}	-101			0	1
	q_{14}	-011			0	0
	q_{15}	-111			0	1

5-15/a ábra

Soros üzemmódu, négy bites kódszavakat vizsgáló ciklikus működésű hibajelző hálózat állapot táblázata

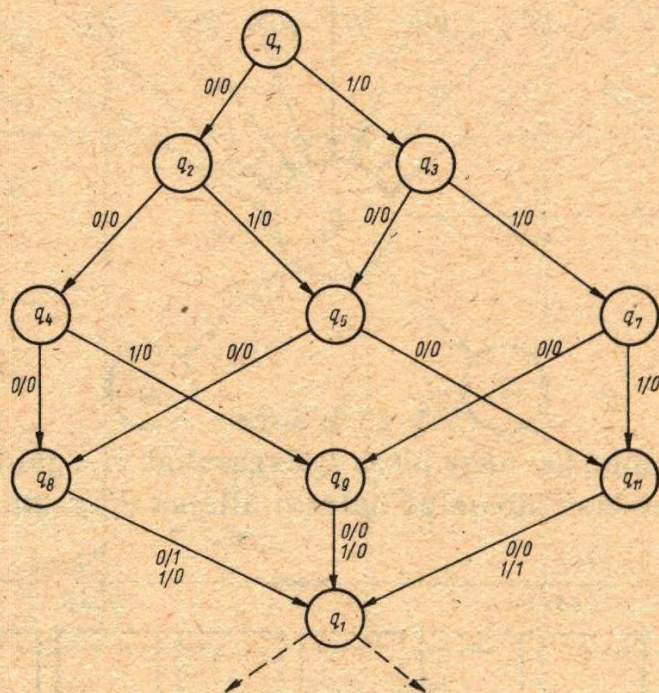


Soros üzemmódu, négy bites kódszavakat vizsgáló ciklikus működésű hibajelző hálózat állapot diagramja



5-15/c ábra
Hibajelző hálózat idődiagramja

k-adik ütem belső állapota	(k+1)-edik ütem belső állapota		Kimenet Y^k	
	$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
q_1	q_2	q_3	0	0
q_2	q_4	q_5	0	0
q_3	q_5	q_7	0	0
q_4	q_8	q_9	0	0
q_5	q_8	q_{11}	0	0
q_7	q_9	q_{11}	0	0
q_8	q_1	q_1	1	0
q_9	q_1	q_1	0	0
q_{11}	q_1	q_1	0	1



5-16. ábra

Egyszerűsített állapot táblázat ill. állapot diagram

b) Felvett állapot táblázat egyszerűsítése

Az állapot táblázat egyszerűsítése a táblázat vizsgálata alapján egyszerű következtetéssel is elvégezhető. Az ekvivalencia mindhárom feltétele teljesül:

$$q_8 \equiv q_{10} \equiv q_{12} = (q_8, q_{10}, q_{12})$$

$$q_{11} \equiv q_{13} \equiv q_{15}$$

$$q_9 \equiv q_{14}$$

csoportokra. A $q_{10} \equiv q_{12}$ ill. $q_{11} \equiv q_{13}$ feltételek maguk után vonják a $q_5 \equiv q_6$ feltétel teljesülését is. A nem egyeztethető elemek: q_1, q_2, q_3, q_4, q_7 . Az egyeztethető csoportokra a következő jelöléseket bevezetve:

$$q_5 = (q_5, q_6)$$

$$q_8 = (q_8, q_{10}, q_{12})$$

$$q_9 = (q_9, q_{14})$$

$$q_{11} = (q_{11}, q_{12}, q_{15})$$

a feladat megoldásához szükséges belső állapotok: $q_1, q_2, q_3, q_4, q_5, q_7, q_8, q_9, q_{11}$.

Az egyszerűsített állapot táblázat és állapot diagram könnyen felrajzolható (5-16. ábra).

Állapot kódolás, a kódolt, egyszerűsített állapot táblázat elkészítése után a feladat az eddig ismertetett módon különösebb nehézség nélkül megoldható.

5.4.3. Hibajelző BCD kód vizsgálatára

Az előző feladatnál bemutatott elvet minden olyan feladat megoldásánál alkalmazhatjuk, ahol 4 bites kódszavak vizsgálatát kell elvégeznünk, és a 4. beérkező impulzus után a rendszer minden esetben alaphelyzetbe tér vissza.

Legyen a feladatunk soros üzemmódban érkező binárisan kódolt decimális kódolású számok ellenőrzése. A beérkező kódszó hibás, ha 10, 11, 12, 13, 14, 15 decimális számoknak megfelelő 1010, 1011, 1100, 1101, 1110, 1111 kódszavak érkeznek be.

a) Belső állapotok felvétele

A belső állapotokat az 5.4.2. pontban leírt gondolatmenet alapján vesszük fel ismét. A k -adik és a $(k+1)$ -edik ütemhez tartozó belső állapotok változatlanok, pusztán a kimenet (hibajelzés) feltétele változik meg az 5-15. ábrához képest. Az 5-17. ábrán felrajzolt állapot táblázat könnyen ellenőrizhető.

Beérkezett impulzusok száma	k -adik ütem belső állapota	Regiszter állapota	$(k+1)$ -edik ütem belső állapota		Kimenet y^k	
			$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
0	q_1	----	q_2	q_3	0	0
1	q_2	---0	q_4	q_5	0	0
	q_3	---1	q_6	q_7		
2	q_4	--00	q_8	q_9	0	0
	q_5	--10	q_{10}	q_{11}		
	q_6	--01	q_{12}	q_{13}		
	q_7	--11	q_{14}	q_{15}		
3	q_8	-000	q_1	q_1	0	0
	q_9	-100			0	1
	q_{10}	-010			0	1
	q_{11}	-110			0	1
	q_{12}	-001			0	0
	q_{13}	-101			0	1
	q_{14}	-011			0	1
	q_{15}	-111			0	1

5-17. ábra

BCD kódban beérkező kódszavak vizsgálatát végző hálózat állapot táblázata

b) Felvett állapot táblázat egyszerűsítése

A felvett állapot táblázat egyszerűsítését részletesen nem végezzük el, csak az egyszerűsítés eredményét közöljük. Közvetlen szemlélet alapján is belátható, hogy

$$(q_9, q_{10}, q_{11}, q_{13}, q_{14}, q_{15}) = q_9$$

$$(q_5, q_7) = q_5$$

$$(q_4, q_6) = q_4$$

$$(q_2, q_3) = q_2$$

$$(q_8, q_{12}) = q_8$$

csoportok összevonhatók. A megmaradt elemek $q_1, q_2, q_4, q_5, q_8, q_9$.

c) Állapot kódolás

A $q_1, q_2, q_4, q_5, q_8, q_9$ (hat darab) belső állapot megvalósításához három tárolót kell (legalább) használni. Az A, B, C tárolók állapotát a hat belső állapothoz rendeljük hozzá pl. a következőképpen:

Belső állapot	A	B	C
q_1	0	0	1
q_2	1	0	1
q_4	1	1	0
q_5	1	0	0
q_8	1	1	1
q_9	0	1	1

d) Realizálás

A felvett állapot kód alapján a kódolt állapot táblázat és igazságtáblázat (5-18. ábra) felhasználásával megrajzoljuk a szekunder változók Veitch diagramjait (5-19. ábra). Kiolvasva:

$$A^{k+1} = (\overline{B}C + B\overline{C}X)^k = A^k (\overline{B}C + B\overline{C}X)^k + (\overline{A}B)^k$$

$$B^{k+1} = (\overline{C} + A\overline{B}X)^k = (\overline{B}C)^k + \overline{B}^k (\overline{C} + A\overline{B}X)^k$$

$$C^{k+1} = (\overline{A} + B + \overline{C})^k = (\overline{A} + B)^k C^k + \overline{C}^k$$

A Veitch diagram segítségével történő minimalizáláshoz felhasználtuk a 0, 2, 8, 10 indexű elő nem forduló mintermeket is.

$(ABC)^k$		$(ABC)^{k+1}$		Y^k	
		$X^k=0$	$X^k=1$	$X^k=0$	$X^k=1$
q_1	001	101	101	0	0
q_2	101	110	100	0	0
q_4	110	111	011	0	0
q_5	100	011	011	0	0
q_8	111	001	001	0	0
q_{11}	011	001	001	0	1

index	k-adik ütem				(k+1)-edik ütem			Y^k
	X^k	A^k	B^k	C^k	A^{k+1}	B^{k+1}	C^{k+1}	
1	0	0	0	1	1	0	1	0
5	0	1	0	1	1	1	0	0
6	0	1	1	0	1	1	1	0
4	0	1	0	0	0	1	1	0
7	0	1	1	1	0	0	1	0
3	0	0	1	1	0	0	1	0
9	1	0	0	1	1	0	1	0
13	1	1	0	1	1	0	0	0
14	1	1	1	0	0	1	1	0
12	1	1	0	0	0	1	1	0
15	1	1	1	1	0	0	1	0
11	1	0	1	1	0	0	1	1

5-18. ábra
(a) Kódolt állapot táblázat,
(b) igazság táblázat

A^{k+1}		B	0	0	1	1
		C	0	1	1	0
X	A					
0	0	x	1			x
0	1		1			
1	1		1			
1	0	x	1			x

$$A^{k+1} = (\bar{B}C + B\bar{C}\bar{X})^k$$

B^{k+1}		B	0	0	1	1
		C	0	1	1	0
X	A					
0	0	x				x
0	1	1	1			1
1	1	1				1
1	0	x				x

$$B^{k+1} = (\bar{C} + A\bar{B}\bar{X})^k$$

C^{k+1}		B	0	0	1	1
		C	0	1	1	0
X	A					
0	0	x	1	1		x
0	1	1		1		1
1	1	1		1		1
1	0	x	1	1		x

$$C^{k+1} = (\bar{A} \cdot B + \bar{C})^k$$

5-19. ábra
Veitch diagramok

A vezérlési függvények:

$$C_{1A} = (\overline{B}C + B\overline{C}X)^k \quad \text{és} \quad C_{2A} = \overline{B}^k$$

$$C_{1B} = \overline{C}^k \quad \text{és} \quad C_{2B} = (\overline{C} + A\overline{X})^k$$

$$C_{1C} = (\overline{A} + B)^k \quad \text{és} \quad C_{2C} = 1$$

A három tárolót pl. JK típusu elemnek realizálva a tárolók bemenetének vezérlésére szolgáló logikai egyenletek:

$$K_A = \overline{C}_{1A} = (BC + \overline{B}\overline{C} + BX)^k$$

$$J_A = C_{2A} = \overline{B}^k$$

$$K_B = \overline{C}_{1B} = C^k$$

$$J_B = C_{2B} = (\overline{C} + A\overline{X})^k$$

$$K_C = \overline{C}_{1C} = (A\overline{B})^k$$

$$J_C = C_{2C} = 1$$

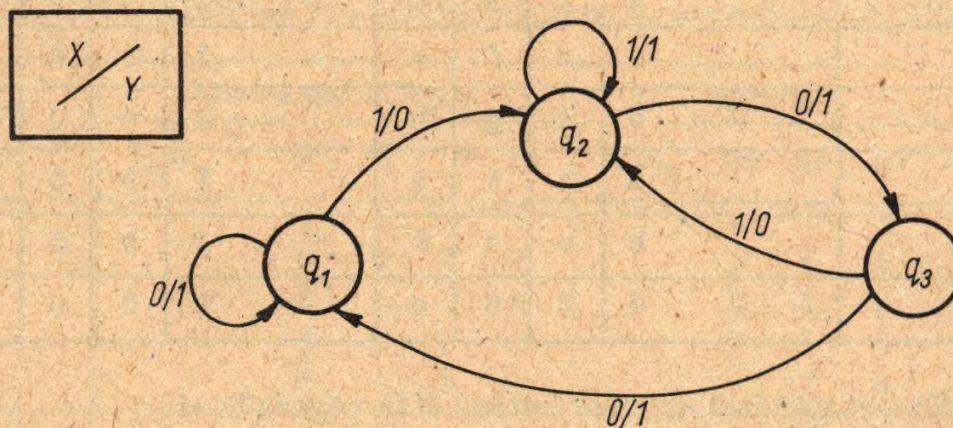
A kimenet (hibajel) logikai egyenlete a redundáns mintermek közül a 10-es indexével összevonva:

$$Y^k = (\overline{A}BX)^k$$

Az állapot kódolás természetesen másképp is elkészíthető. Az állapot kódolás lényegesen befolyásolja a vezérlési függvényeket, és a kombinációs hálózat felépítését. A feladatot általában minimális számú alkatrész felhasználásával kívánjuk megoldani. A minimális alkatrész számot eredményező állapot kódolás meghatározása partició segítségével történhet.

5.5 Gyakorló feladatok

1. Határozzuk meg az 5-20. ábrán látható állapot diagrammal rendelkező szekvenciális hálózat állapot táblázatát és igazságtáblázatát.



5-20. ábra

2. Egyszerűsítsük az alábbi állapot táblázatokat

- csoportokra bontással (HUFFMAN-MEALY módszerrel)
- Lépcsős táblázatok segítségével.

2.1.

Kiinduló állapot	Következő állapot		Kimenet	
	X=0	X=1	X=0	X=1
a	b	c	0	0
b	a	c	1	0
c	e	f	0	1
d	b	d	0	0
e	c	f	1	1
f	b	e	0	0

2.2.

Kiinduló állapot	Következő állapot					Kimenet				
	X ₁ X ₂	00	01	10	11	X ₁ X ₂	00	01	10	11
a		a	c	c	f		1	0	0	0
b		b	e	a	f		1	0	0	1
c		a	d	f	e		1	0	0	0
d		d	c	g	b		1	0	0	0
e		d	a	f	e		1	0	0	0
f		b	c	a	f		1	0	0	1
g		d	d	b	e		1	0	0	0

Az állapot táblázat egyszerűsítése után végezzük el:

- állapot kódolást
- realizáljuk a hálózatot.

3. Egy szekvenciális hálózatot három tárolóval (A, B, C) realizálva a tárolók vezérlésére szolgáló logikai egyenletek a két bemeneti változó (X_1, X_2) felhasználásával:

$$J_A = \overline{B}X_2 + B\overline{X}_2;$$

$$K_A = X_1\overline{X}_2\overline{B}C + \overline{X}_1\overline{X}_2\overline{B}$$

$$J_B = \overline{X}_2 + \overline{A}$$

$$K_B = \overline{X}_2\overline{C} + X_2C + \overline{X}_1\overline{A}$$

$$J_C = B\overline{X}_2 + A\overline{X}_1 + \overline{A}BX_1;$$

$$K_C = \overline{X}_1 + \overline{X}_2$$

A kimenet (Y) logikai egyenlete:

$$Y = B\overline{X}_2 + A\overline{X}_1\overline{X}_2 + \overline{A}X_1X_2 + C\overline{X}_1X_2$$

Az egyszerűsítésre felhasználható redundáns termek:

$$\overline{X}_1BC = \overline{X}_1\overline{B}C = X_1\overline{A}B = X_1\overline{B}C = 0$$

Kérdés: megvalósítható-e a feladat követelményeit kielégítő szekvenciális hálózat két tároló segítségével? Ha igen, akkor írjuk fel a két tároló bemeneti egyenleteit és a kimenet (Y) logikai egyenletét.

4. Az alábbi állapot táblázatban határozatlan állapotok is szerepelnek, mégpedig

- a) a következő belső állapot oszlopában
- b) kimenet oszlopaiban.

A határozatlan (tetszőleges) belső állapotokat bármely belső állapottal helyettesíthetjük. A kimenet oszlopaiban szereplő határozatlan állapotok értéke 0 vagy 1 lehet. A határozatlan állapotokat vízszintes egyenesekkel jelöljük.

Kiinduló állapot	Következő állapot					Kimenet				
	$X_1 X_2$	00	01	10	11	$X_1 X_2$	00	01	10	11
a		a	b	d	c		0	1	0	0
b		d	a	b	c		0	1	0	1
c		c	b	b	c		0	1	0	0
d		d	-	b	c		-	1	0	-

Egyszerűsítsük az állapot táblázatot.

5. Tervezendő egy szekvenciális hálózat, amely két bemenettel (X_1, X_2) és egy kimenettel (Y) rendelkezik. A bemenetre érkező jelek ($X_1 X_2$) értéke 00, 01, 10, 11. A kimeneten akkor jelenik meg 1-es jel, ha a beérkezett utolsó négy vagy annál nagyobb számú $X_1 X_2$ impulzus megegyezik, vagy különbözik. Ha tehát az utolsó négy vagy több impulzusra

$$X_1 \oplus X_2 = 1, \text{ vagy } X_1 \odot X_2 = 1$$

akkor $Y = 1$. Az áramkör az első $X_1 X_2$ impulzuspár beérkezésekor az "a" állapotban van. Az áramkör működését a következő példa szemlélteti:

X_1	0	0	1	0	0	1	0	0	1	0	1	0	0	1	1	1	0
X_2	1	1	0	0	1	1	0	0	1	0	1	1	1	0	0	0	0
Belső állapot	a	c	f	h	b	e	b	d	g	g	g	g	c	f	h	h	h
Kimenet (Y)	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0

A szekvenciális hálózat állapot táblázata:

Kiinduló állapot	Következő állapot					Kimenet				
	X ₁ X ₂	00	01	10	11	X ₁ X ₂	00	01	10	11
a			c					0		
b		d	e				0	0		
c			f					0		
d		g					0			
e					b					0
f				h					0	
g		g	c		g		1	0		1
h		b		h			0		1	

Rajzoljuk meg a hálózat állapot diagramját. Egyszerűsítsük az állapot táblázatot. Realizáljuk a szekvenciális hálózatot.

6. Oldjuk meg az előző feladatot arra az esetre, amikor a kimeneten csak akkor jelenik meg logikai Igen szint, ha

- az utolsó négy vagy több impulzus mindkét bemenő vezetéken megegyezik
- az utolsó négy vagy több impulzus mindkét bemenő vezetéken különbözik.

7. Tervezzünk szekvenciális hálózatot, mely egy bemenettel (X) és egyetlen kimenettel (Y) rendelkezik. A kimenet logikai Igen szint jelenik meg, ha az utolsó 4 vagy több bemenő impulzus értéke 1.

A beérkezett utolsó három bitet tárolva a hálózat állapot táblázatát a következőképpen rajzolhatjuk fel:

Kiinduló állapot	Következő állapot		Kimenet (Y)	
	X=0	X=1	X=0	X=1
$q_0(000)$	q_0	q_1	0	0
$q_1(001)$	q_2	q_3	0	0
$q_2(010)$	q_4	q_5	0	0
$q_3(011)$	q_6	q_7	0	0
$q_4(100)$	q_0	q_1	0	0
$q_5(101)$	q_2	q_3	0	0
$q_6(110)$	q_4	q_5	0	0
$q_7(111)$	q_6	q_7	0	1

Egyszerűsítsük az állapot táblázatot csoportokra bontással. Realizáljuk a szekvenciális hálózatot.

8. Soros üzemi számítógép 5-ből 2 kódban dolgozik. A decimális számjegyeknek 5-ből 2 kódját az alábbi táblázat tartalmazza:

Decimális számjegy	t_5	t_4	t_3	t_2	t_1
0	0	0	0	1	1
1	0	0	1	0	1
2	0	1	0	0	1
3	1	0	0	0	1
4	0	0	1	1	0
5	0	1	0	1	0
6	1	0	0	1	0
7	0	1	1	0	0
8	1	0	1	0	0
9	1	1	0	0	0

A soros kód legalacsonyabb helyértéken álló számjegye t_1 időpontban, legmagasabb helyértéken álló számjegye t_5 időpontban érkezik.

Tervezzük meg azt a szekvenciális hálózatot, amely t_5 időpontban jelzést ad, ha a kódszó hibás, vagyis a kódszóban több vagy kevesebb 1-es van, mint 2. A hálózat kimenő jele a t_1, t_2, t_3, t_4 időpontban mindig zérus.

9. A "BCD" kód hibajelzésének vizsgálatánál bemutatott elven tervezzünk egy párosság ellenőrző hálózatot. A bemenő 4 bites kódszavakat megvizsgálja az áramkör, és a 4. beérkező impulzus után a rendszer mindig alaphelyzetbe tér vissza.

A beérkező kódszó hibás, ha a benne levő egyesek száma páratlan. Ebben az esetben az Y kimeneten hibajel jelenik meg.

A szekvenciális hálózat állapot táblázata az 5.4.3. pontban bemutatott elven közvetlenül felrajzolható, csak a kimenetek oszlopában jelentkezik eltérés.

Rajzoljuk fel a szekvenciális hálózat állapot diagramját, állapot táblázatát. Realizáljuk a hálózatot.

6. FÜGGELÉK

Függelékben közöljük a jegyzetben használt minterm ill. maxterm táblákat és összefoglaljuk logikai függvények kétszintes realizálásának különböző lehetőségeit.

6.1 Minterm tábla

Ha a függvény diszjunktív normál alakban adott, a minterm táblán úgy ábrázoljuk, hogy ahol $f_i = 1$, ott a megfelelő négyzetbe 1-et írunk. A 0 együtt-hatóju mintermeket külön nem tüntetjük fel, ezért ahol a minterm táblában nincs 1-jel, ott a mintermek együtt-hatója megállapodásszerűen 0.

Többváltozós függvények minterm tábláit a 6-1. ábrán láthatjuk. Az nyilvánvaló, hogy az egyes mintermeket $(2^n)!$ módon helyezhetjük el a táblán. Mi egy olyan elrendezést választottunk ki, mely lehetővé teszi a sorok ill. oszlopok ellentétes végén levő mintermek esetleges összevonását. Tehát a felrajzolt táblák a széleken összefüggnek. A legutolsó sor (oszlop) után az első sor (oszlop) következik. Természetesen nem ez az egyetlen elrendezés, amely ezt a követelményt kielégíti.

6.2 Maxterm tábla

$$F = \sum_{i=0}^{2^n-1} f_i m_i^n = \prod_{i=0}^{2^n-1} (f_i + M_{2^n-1-i}^n)$$

összefüggés alapján láthatjuk, hogy a szorzatban csak azok a maxtermek szerepelnek, melyekhez tartozó f_i értéke zérus. Az ezekhez tartozó maxterm:

$$M_{2^n-1-i}^n = m_i^n$$

A 0 1

0	1
---	---

n=1

A

B 0 1

0	0	1
1	2	3

n=2

B

B 0 0 1 1
C 0 1 1 0

0	0	1	3	2
1	4	5	7	6

n=3

B

C

C 0 0 1 1
D 0 1 1 0

0 0	0	1	3	2
0 1	4	5	7	6
1 1	12	13	15	14
1 0	8	9	11	10

n=4

C

D

6-1/a ábra
Többváltozós függvények minterm táblái

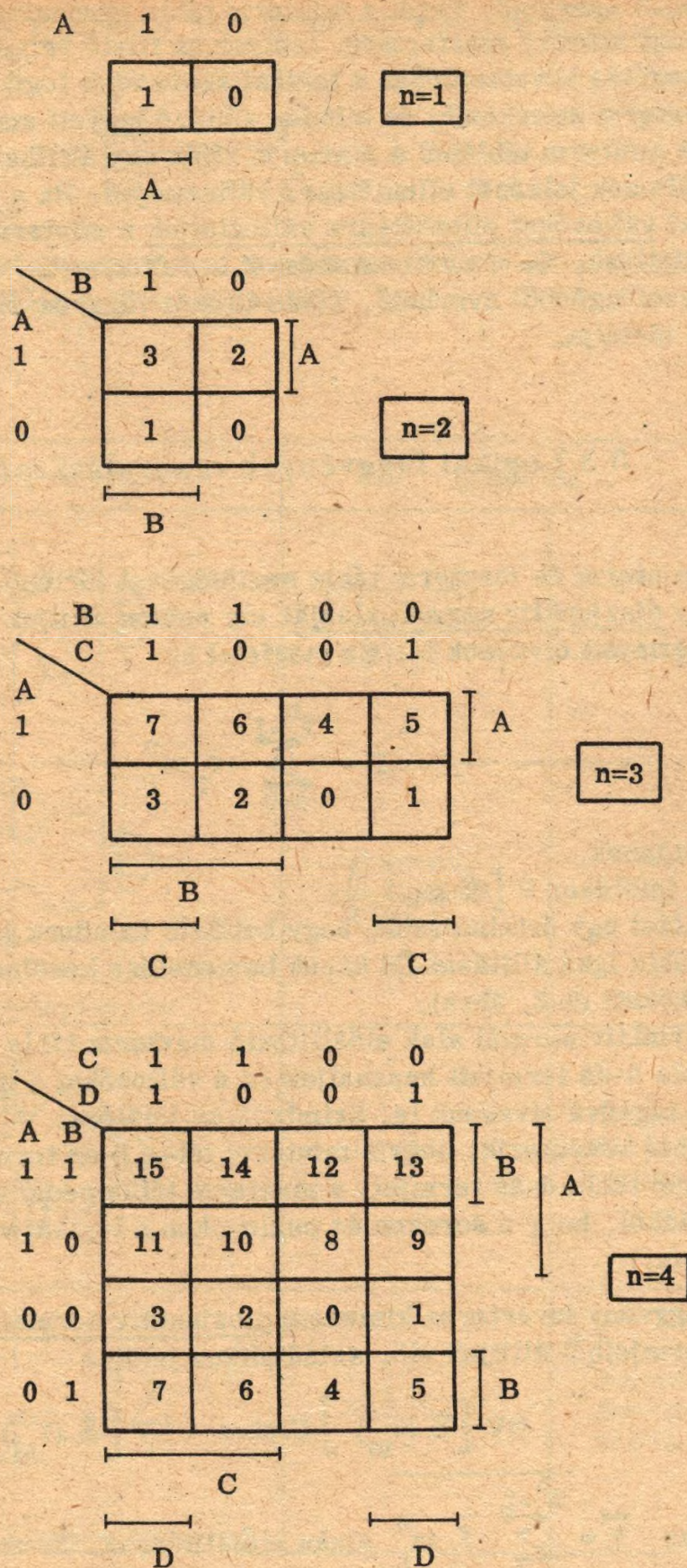
				C					
D				D					
A	0	1	3	2	4	5	7	6	B
	8	9	11	10	12	13	15	14	
	24	25	27	26	28	29	31	30	
	16	17	19	18	20	21	23	22	
E				E					

n=5

								D	
E				E					
A	0	1	3	2	4	5	7	6	C
	8	9	11	10	12	13	15	14	
	24	25	27	26	28	29	31	30	
	16	17	19	18	20	21	23	22	
	32	33	35	34	36	37	39	38	B
	40	41	43	42	44	45	47	46	
	56	57	59	58	60	61	63	62	
	48	49	51	56	52	53	55	54	
F				F					

n=6

6-1/b ábra
Többváltozós függvények minterm táblái



6-2. ábra
Többváltozós függvények maxterm táblái

összefüggésből nyerhető. Tehát a minterm tábla zérusainak helyén állnak a függvényt meghatározó maxtermek, melyek az illető helyen álló minterm negáltjai. A negálás következtében a logikai szorzatból logikai összeg lesz (mintermből maxterm keletkezik) és minden változó helyett annak tagadott értéke szerepel. A minterm táblából a maxterm tábla úgy állítható elő, hogy a tábla szélein a változók jelzését ellentétesre változtatjuk. Ha a sorokon és oszlopokon a logikai változókat ellentétesre változtatjuk a minterm táblából maxterm táblát készítettünk. Ha a maxterm indexét is feltüntetjük, akkor az új index $(2^n - 1 - i)$ összefüggésből nyerhető. Többváltozós függvények maxterm tábláit a 6-2. ábra mutatja.

6.3 Logikai függvények realizálási módjai

a) A minterm és maxterm tábla segítségével történő minimalizálás után egy függvény diszjunktív normál alakját oly módon állítjuk elő, hogy a minterm tábla 1-es termjeit olvassuk le. Ez megfelel az

$$F = \sum_{i=0}^{2^n-1} f_i m_i^n$$

alaku előállításnak.

Szimbolikus jelölése: $V [\bar{E} (1_m)]$

A képletet úgy értelmezzük, hogy belülről haladunk jobbról balra. Vagyis a minterm tábla igaz állításai ÉS kapuk bemenetére kerülnek az ÉS kapukat VAGY kapu követi (6-3. ábra).

A diszjunktív normál alak előállítható maxterm tábla segítségével, ha a maxterm tábla 0-ás termjeit használjuk és a változókat a maxterm tábla sorain és oszlopain tagadva olvassuk le. Szimbolikus jelölése: $V [\bar{E} (0_M)]$.

(Ez abból következik, hogy a minterm tábla 1-es termjeinek helyén állnak a maxterm tábla 0-ás termjei, a maxterm tábla pedig úgy származik a minterm táblából, hogy a sorokon és oszlopokon a logikai változókat ellentétesre változtatjuk.)

b) A függvény inverteres kimenetű diszjunktív normál alakját a minterm tábla 0-ás termjeiből állítjuk elő, szimbolikus jelölés

$$NV [\bar{E} (0_m)] \text{ illetve } NV [\bar{E} (1_M)]$$

ez megfelel az $\bar{F} = \sum_{i=0}^{2^n-1} \bar{f}_i m_i^n$ alaku előállításnak. Ez az eredeti függvény

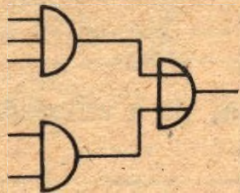
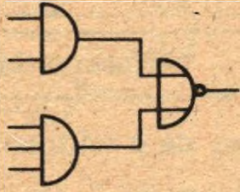
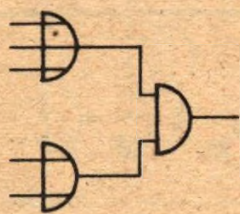
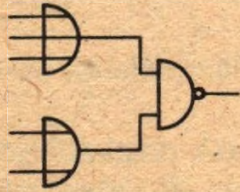
diszjunktív konjunkciós alakját adja inverteres kimenettel.

A tagadott függvényben - mint már láttuk - azok a mintermek szerepelnek, melyek a függvény igaz értékéből hiányoznak.

c) Egy függvény konjunktív normál alakját a maxterm tábla 1-es termjeiből állítjuk elő. Ez az

$$F = \prod_{i=0}^{2^n-1} (f_i + M_{2^n-1-i}^n)$$

alaku előállításnak felel meg, mivel a minterm tábla 1-es termjeinek helyébe ($f_i = 1$) a maxterm táblán 0-át írunk.

	Logikai vázlat	Előállítása	
Diszjunktív normál alak		Minterm tábla, 1-es termjeiből $V [\dot{E}(1_m)]$	$V [\dot{E}(0_m)]$
Diszjunktív konjunkciós alak inverteres kimenettel	 Kimeneten inverter	Minterm tábla 0-ás termjeiből $NV [\dot{E}(0_m)]$	$NV [\dot{E}(1_m)]$
Konjunktív normál alak		$\dot{E} [V(0_m)]$	Maxterm tábla 1-es termjeiből $\dot{E} [V(1_m)]$
Konjunktív diszjunkciós alak inverteres kimenettel	 Kimeneten inverter	$N\dot{E} [V(1_m)]$	Maxterm tábla 0-ás termjeiből $N\dot{E} [V(0_m)]$

6-3. ábra
Diszjunktív és konjunktív normál alak meghatározása
Veitch diagram segítségével

Szimbólikus jelöléssel: $\bar{E}[V(1_M)]$, vagy a minterm táblát használva $\bar{E}[V(\bar{0}_m)]$. Tehát a minterm tábla 0-ás termjeit vesszük (ez megfelel a maxterm tábla 1-es termjeinek) és a változókat tagadva olvassuk le (mivel a minterm táblából a maxterm tábla úgy nyerhető, hogy a minterm táblában a sorokon és oszlopokon a logikai változókat ellentétesre változtatjuk).

d) A függvény inverteres kimenetű konjunktív normál alakját a maxterm tábla 0-ás termjeiből, vagy a minterm tábla 1-es termjeiből a sorokat és oszlopokat tagadva olvasva le állítjuk elő. Szimbólikus jelölés: $\bar{N}\bar{E}[V(0_M)]$ ill. $\bar{N}\bar{E}[V(\bar{1}_m)]$.

Ez az eredeti függvény konjunktív diszjunkciós alakját adja inverteres kimenettel (6-3. ábra).

Az eddigiekből megjegyezhetjük a következő szabályt:

Minterm tábla	Maxterm tábla
1-es termjének megfelel	0-ás termje, de a változókat tagadva olvassuk le
0-ás termjének megfelel	1-es termje, de a változókat tagadva olvassuk le

Maxterm tábla	Minterm tábla
1-es termjének megfelel	0-ás termje, de a változókat tagadva olvassuk le
0-ás termjének megfelel	1-es termje, de a változókat tagadva olvassuk le

Szimbólikus jelöléssel:

$$1_m \doteq \bar{0}_M \quad \text{ill.} \quad 0_m \doteq \bar{1}_M$$

és

$$1_M \doteq \bar{0}_m \quad \text{ill.} \quad 0_M \doteq \bar{1}_m$$

A külső (kimeneten levő) függvény tagadása esetén az $1 \rightarrow 0$ ill. $0 \rightarrow 1$ cserét kell alkalmazni.

Még további négy realizálási módot fogunk meghatározni. A belső (a beemeneten levő) logikai függvény eddig konjunktív ill. diszjunktió volt.

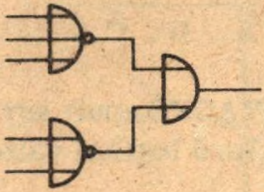
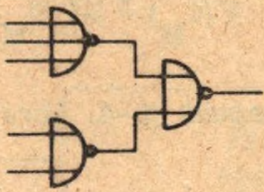
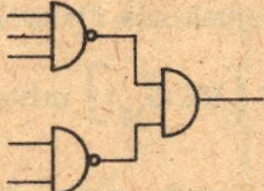
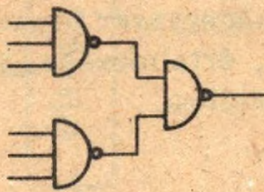
Válasszuk ki az egyik konjunktív tagot. Legyen ez pl. $\bar{A}\bar{B}\bar{C}$, kétszer tagadva:

$$\bar{A}\bar{B}\bar{C} = \overline{\overline{\bar{A}\bar{B}\bar{C}}} = \overline{A + B + C}$$

Bármely belső konjunkciós tagot (ÉS kaput) egy VAGY kapuval helyettesíthetjük, de a bemenetén levő logikai változókat tagadnunk kell. Szimbolikus írásmódban:

$$\bar{E}(1_m) \doteq NV(\bar{1}_m) \quad \text{ill.} \quad \bar{E}(0_m) \doteq NV(\bar{0}_m)$$

Ily módon két új realizálási módot kaptunk (6-4. ábra).

	Logikai vázlat	Előállítás	
Kettős diszjunkció		$V[NV(\bar{1}_m)]$	Maxterm tábla 0-ás termjeiből $V[NV(0_m)]$
Kettős diszjunkció inverteres kimenettel	 Peirce - f. művelet	$NV[NV(\bar{0}_m)]$	Maxterm tábla 1-es termjeiből $NV[NV(1_m)]$
Kettős konjunkció		Minterm tábla 0-ás termjeiből $\bar{E}[\bar{N}E(0_m)]$	$\bar{E}[\bar{N}E(\bar{0}_m)]$
Kettős konjunkció inverteres kimenettel	 Sheffer - f. művelet	Minterm tábla 1-es termjeiből $\bar{N}E[\bar{N}E(1_m)]$	$\bar{N}E[\bar{N}E(\bar{0}_m)]$

6-4. ábra
Realizálási módok meghatározása Veitch diagram segítségével

A diszjunktív normál alakból:

$$V[\bar{E}(1_m)] \rightarrow V[NV(\bar{1}_m)] \text{ ill. } V[NV(0_M)]$$

Az inverteres kimenetű diszjunktív konjunkciós alakból kiindulva:

$$NV[\bar{E}(0_m)] \rightarrow NV[NV(\bar{0}_m)] \text{ ill. } NV[NV(1_M)]$$

mivel $\bar{0}_m \doteq 1_M$.

Végül válasszuk ki a bemeneten levő egyik diszjunktív tagot. Legyen ez pl. $A + \bar{B} + C$. Kétszer tagadva:

$$A + \bar{B} + C = \overline{\overline{A + \bar{B} + C}} = \overline{\bar{A} \cdot B \cdot \bar{C}}$$

Igy bármely belső diszjunktív tagot (VAGY kaput) egy NÉS kapuval helyettesíthetünk, de a VAGY kapu bemenetén levő logikai változókat tagadnunk kell. Szimbolikus írásmódban:

$$V(1_M) \doteq N\bar{E}(\bar{1}_M) \text{ ill. } V(0_M) \doteq N\bar{E}(\bar{0}_M)$$

Ily módon további két új realizálási módot kapunk (6-4. ábra).

A konjunktiiv normál alakból:

$$\bar{E}[V(\bar{0}_m)] \rightarrow \bar{E}[N\bar{E}(0_m)] \text{ ill. } \bar{E}[N\bar{E}(\bar{1}_M)] \text{ mivel } 0_m \doteq \bar{1}_M$$

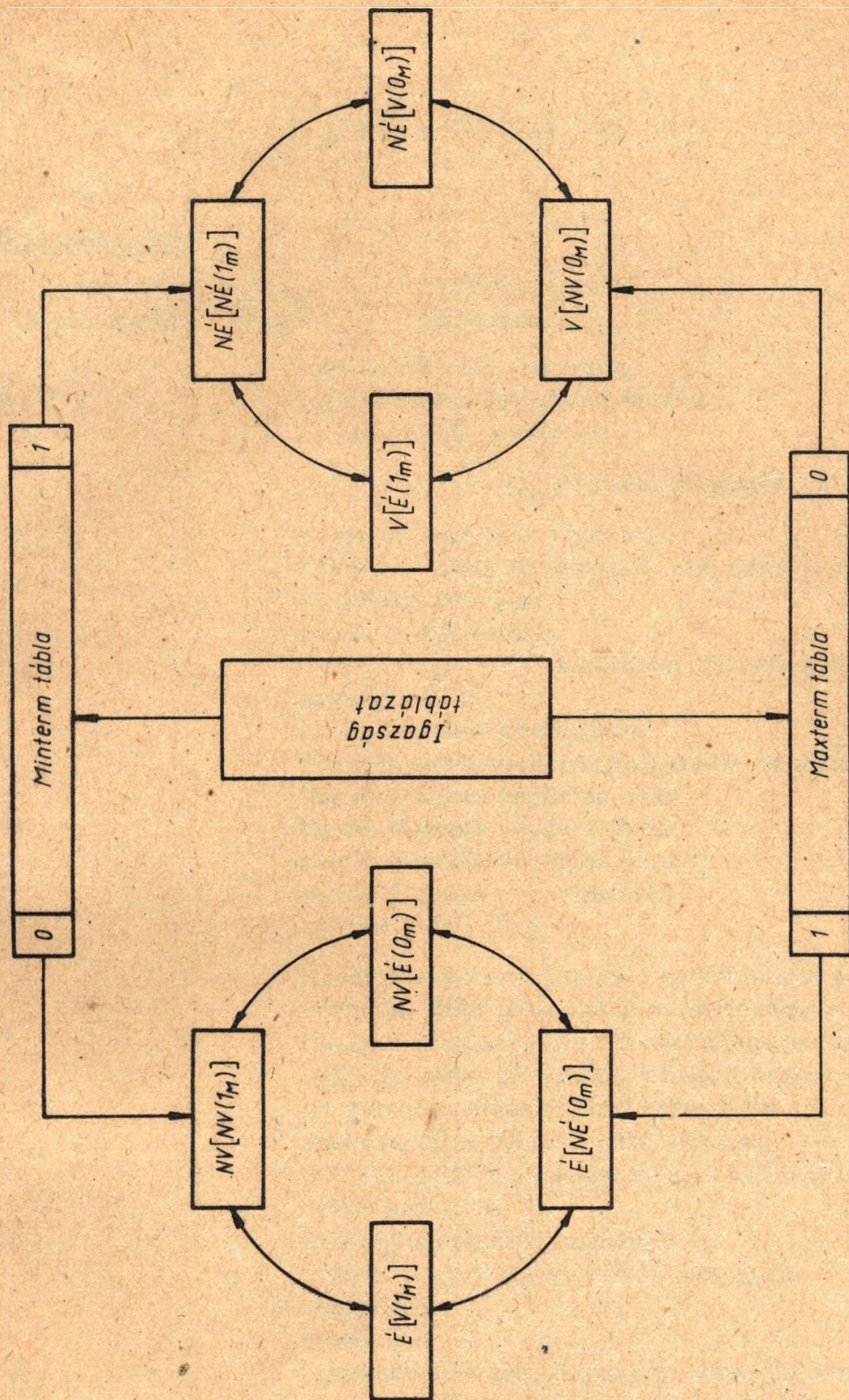
Az inverteres kimenetű konjunktiiv diszjunktív alakból kiindulva:

$$N\bar{E}[V(\bar{1}_m)] \rightarrow N\bar{E}[N\bar{E}(1_m)] \text{ ill. } N\bar{E}[N\bar{E}(\bar{0}_M)] \text{ mivel } 1_m \doteq \bar{0}_M$$

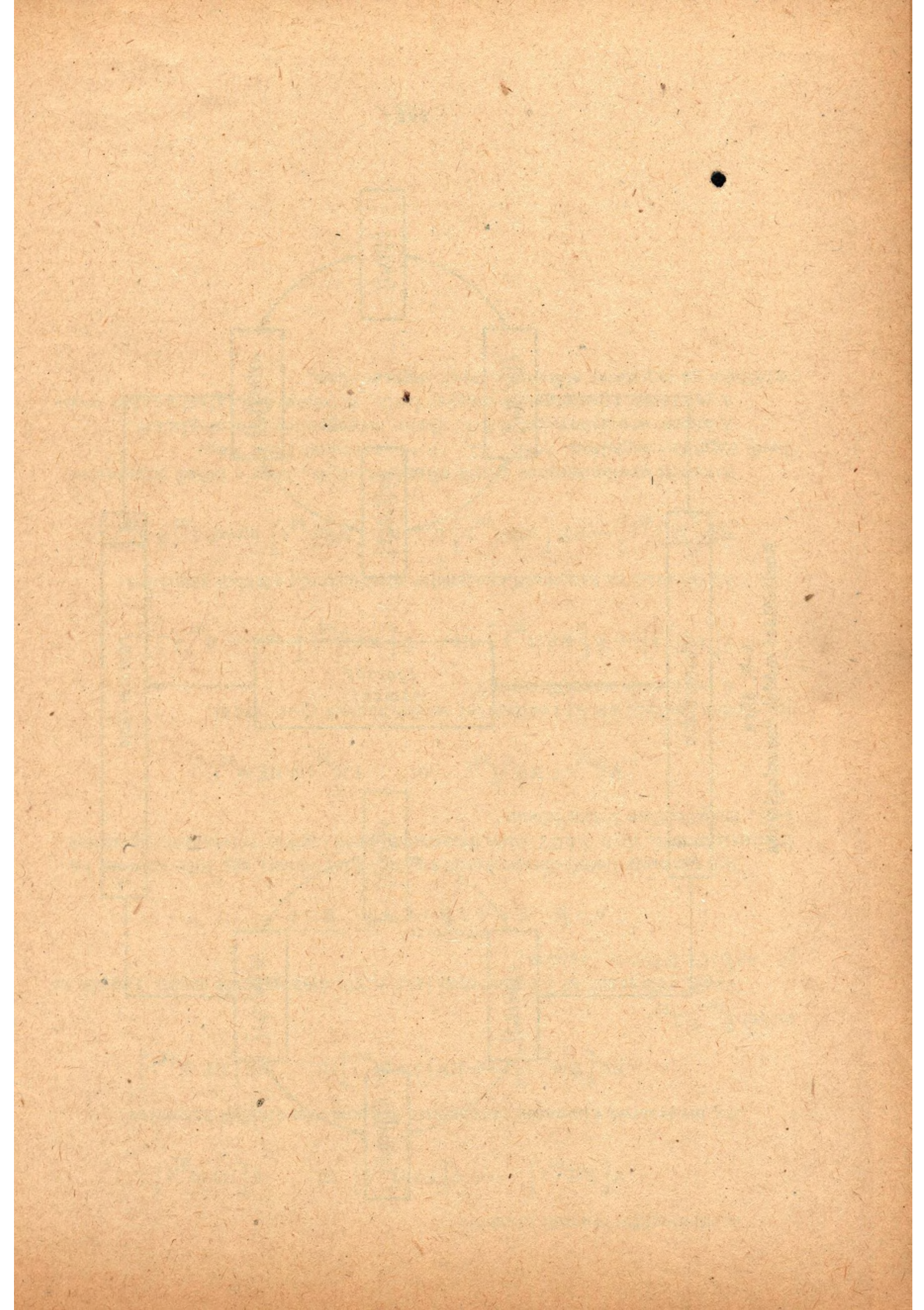
A kapott eredményeken ismét ellenőrizhetjük, hogy a külső (kimeneten levő) függvény tagadása $1 \rightarrow 0$ ill. $0 \rightarrow 1$ cserének felel meg.

A kapott eredményeket a 6-5. ábrán még egyszer összefoglaltuk.

A jelölések részletesebb magyarázata, értelmezése, alkalmazása megtalálható az egyetemi előtanulmányok jegyzeteiben.



6-5. ábra
Realizálási módok összefoglalása



7. JELMAGYARÁZAT

Logikai műveletek

Diszjunkció:	+	Ekvivalencia:	$\odot$
Konjunkció:	pont	Antivalencia:	$\oplus$
$A, B, C \dots$		logikai változók (ítéletek)	
$\bar{A}, \bar{B}, \bar{C} \dots$		logikai változók tagadott értéke	
C_1, C_2		tároló általános alakban	
		$(Q^{k+1} = C_1 Q^k + C_2 \bar{Q}^k)$ felírt állapotegyenletében	
D		szereplő vezérlési függvények	
Δ		késleltető elem (D típusu tároló) bemenete	
Δ^i		késleltető operátor	
f_i		i-szeres késleltetés	
$\underline{I}$		Veitch diagram "i" indexű termjének együtthatója	
J, K		egység mátrix	
$\underline{M}$		JK típusu tároló bemenetei	
$Q(t_0)$		lineáris szekvenciális hálózat állapot mátrixa	
$Q(\tau)$		tároló állapota vezérlés előtt	
Q^k		tároló állapota vezérlés után	
Q^{k+1}		tároló állapota vezérlés előtt	
Q_1		tároló állapota vezérlés után	
Q_1^k		hálózat belső állapota	
$\underline{Q_1^k}$		lineáris szekvenciális hálózat késleltető elemeinek	
$\underline{Q_1^{k+1}}$		vezérlés előtti állapotából alkotott oszlop vektor	
q		lineáris szekvenciális hálózat késleltető elemeinek	
q_i		vezérlés utáni állapotából alkotott oszlop vektor	
R, S		aszinkron működésű tároló kimenete	
		szekvenciális hálózat belső állapota	
R, S, T		RESET, SET bemenettel rendelkező tároló törlő és	
T		beíró bemenete	
τ		RST típusu tároló bemenetei	
V		trigger vagy T típusu tároló bemenete	
V_0, V_1		késleltetés	
		vezérlőjel	
		statikus kapuzású, dinamikus vezérlésű tároló	
		dinamikus bemenetei	

X_i
 Y_i

i-edik bemenő változó

i-edik kimenő változó

"1"-es állapot

valamely tároló "1"-es állapotban van, ha az "1" jelű (Q) kimenetén logikai Igen szint van

"0"-ás állapot

valamely tároló "0"-ás állapotban van, ha a "0" jelű ($\bar{Q}$) kimenetén logikai Igen szint van.

$1 \rightarrow 0$

"1"-es állapotból "0"-ás állapotba történő billenés szimbolikus jelölése. Negatív logikai szintű rendszerekben pozitív impulzust (élt) jelent

$0 \rightarrow 1$

"0"-ás állapotból "1"-es állapotba történő billenés szimbolikus jelölése

$Q^k \rightarrow Q^{k+1}$

regiszter (tároló) vezérlés hatására bekövetkező állapot változása

$V : (A) \rightarrow B$

V vezérlőjel hatására A regiszter tartalma átiródik B regiszterbe

$V : (A_{i-1}) \rightarrow A_i$

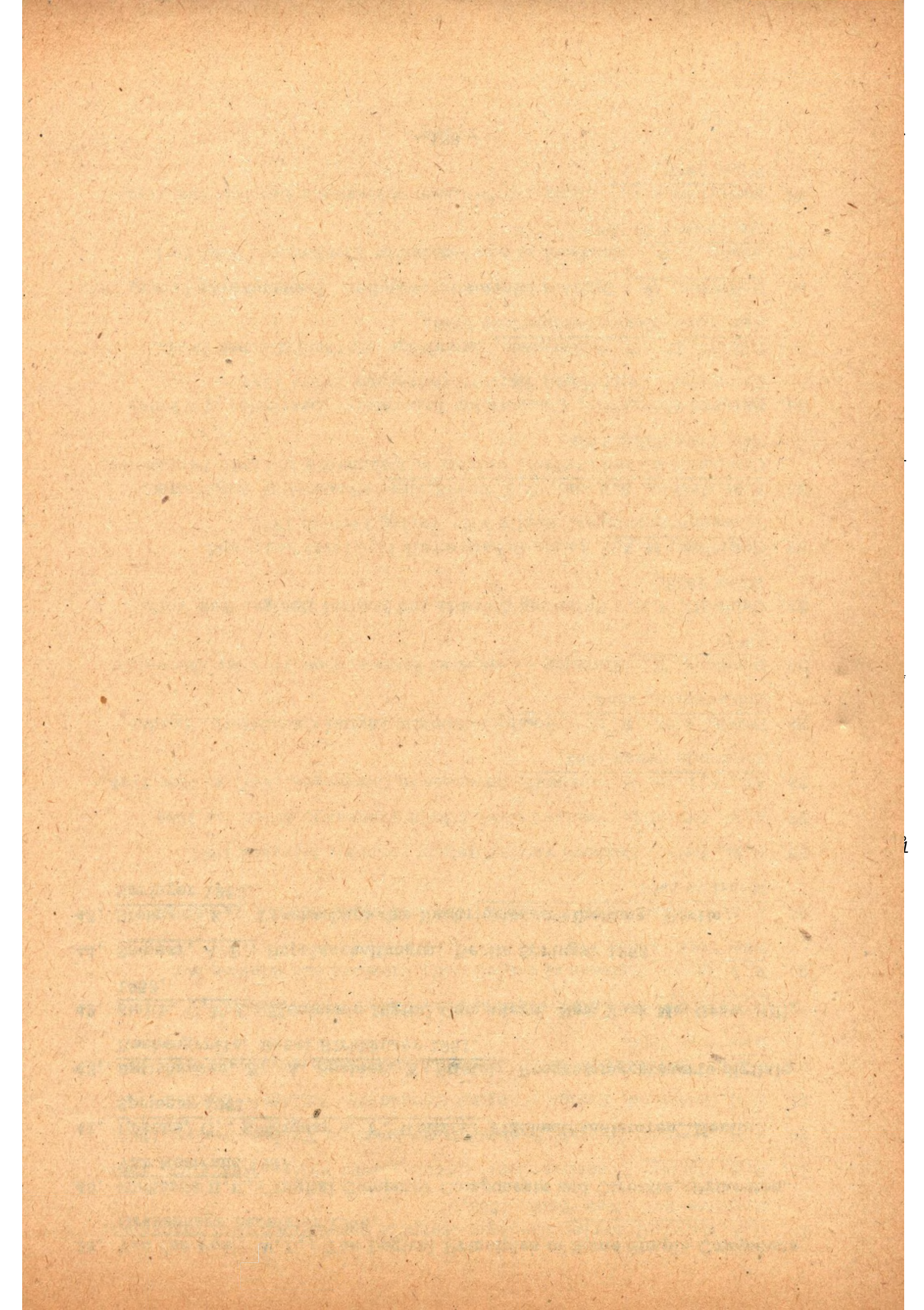
V vezérlőjel hatására az A regiszter (A_{i-1})-edik elemének (cellájának) tartalma átiródik az A regiszter i-edik elemébe (cellájába)

8. IRODALOM

1. dr. Gál J., Szittyá O., Flesch I., Ruppenthal P.: Logikai kapcsolástan. Tankönyvkiadó. 1966.
2. Dr. Csáki Frigyes: Vezérléstechnika. Tankönyvkiadó 1966.
3. Gertler János: Vezérléstechnika. Tankönyvkiadó 1966.
4. Bohus Miklós: Elektronikus számítógépek. Tankönyvkiadó. 1966.
5. Bohus Miklós: Kiegészítés az elektronikus számítógépek c. jegyzethez. Tankönyvkiadó. 1967.
6. Tóth Mihály; Janovics Sándor: Digitális rendszertechnika. Tankönyvkiadó. 1968.
7. dr. Gordos Géza, Varga András: Adatátvitel és adatfeldolgozás. Tankönyvkiadó. 1968.
8. Szittyá Ottó: Digitális vezérléstechnika. Tankönyvkiadó. 1966.
9. Dr. Tarján Rezső: Elektronikus digitális számítógépek. Tankönyvkiadó. 1962.
10. Orbán Miklós: Digitális számítógépek. Tankönyvkiadó. 1967.
11. Dr. Benedikt Szvetlana: Digitális technika. Tankönyvkiadó. 1968.
12. Y. Chu: Digitális számítógépek tervezésének alapjai. Műszaki Könyvkiadó. 1966.
13. M. Phister: Logical design of digital computers. John Wiley. New York. 1958.
14. T.C. Bartee, I.L. Legow, I.S. Reed: Theory and design of digital machines. Mc. Graw Hill, New York. 1962.
15. W. Peterson: Error-correcting codes. John Wiley. New York. 1961.
16. H. Hellerman: Digital computer system principles. Mc. Graw Hill. New York. 1967.
17. I. Flores: The logic of computer arithmetic. Prentice-Hall. London. 1963.

18. G.A. Maley, J. Earle: The logic design of transistor digital computers. Prentice-Hall, New-York. 1963.
19. R.B. Hurley: Transistor logic circuits. John Wiley. New York. 1961.
20. A.P. Speiser: Digitale Rechenanlagen Springer Verlag. Berlin. 1965.
21. H.H. Gschwind: Design of Digital Computers. Springer. Berlin. 1967.
22. A. Gill: Introduction to the Theory of finite-state machines. Mc.Graw-Hill. New York 1962.
23. R.E. Miller: Switching Theory. John Wiley. New York 1965.
24. W.V. Quine: Methods of Logic. Holt, Rinehard and Winston Inc. New York. 1950.
25. E.A. Drozdov, B.I. Prohorov: Osznovü vicsiszlitelnoj tehniki. Moszkva 1964.
26. P.M. Belas: Osznovü vicsiszlitelnoj tehniki, Moszkva 1964.
27. A.M. Szucsilin: Osznovü vicsiszlitelnoj tehniki. Moszkva. 1964.
28. Alt, F.L. u. M. Runinoff: Advances in Computers. Vol. 4, New York, Academic Press 1963.
29. Booth, A.D., K.H.V. Booth: Automatic Digital Calculators. London. Butterworts. 1956.
30. Buchholz, V.: Planning a Computer System. New York Mc.Graw-Hill. 1961.
31. Caldwell, S.H.: Switching Circuits and Logical Design. New York, Wiley 1958.
32. Gawrilow, M.A.: Relais-Schalttechnik für Starkstrom- und Schwachtrömanlagen. Berlin VEB Verlag Technik 1953.
33. E.M. Grabbe, S. Ramo, D.E. Wooldridge: Handbook of Automation, Computation, and Control, Volume 2, Computers and Data Processing, New York Wiley 1959.
34. Harvard University, Computation Laboratory: Desription of a Relay Calculator, Cambridge: Harvard University Press 1949.
35. Hilbert, D., W. Ackermann: Grundzüge der theoretischen Logik. New York. Dover Publications 1946.
36. Hoffmann, W.: Digitale Informationswandler. Braunschweig Vieweg 1962.
37. Hunter /L.P.: Handbook of Semiconductor Electronics, New York. Mc.Graw-Hill 1962.
38. Meyerhoff A.H.: Digital Applications of Magnetic Devices New York: Wiley 1960.

39. Van der Poel, W.L.: The Logical Principles of Some Simple Computers
Gravenhage Excelsior 1956.
40. Richards R.K.: Digital Computer Components and Circuits. Princeton.
Van Nostrand 1957.
41. Rusche, G., K Wagner u. F. Weitzsch: Flächentransistoren. Berlin.
Springer 1961.
42. Rutishauser, H., A. Speiser, E. Stiefel: Programmgesteuerte digitale
Rechengeräte. Basel Birkhäuser 1951.
43. Smith, C.V.L.: Electronic Digital Computers. New York Mc.Graw-Hill.
1959.
44. Speiser, A.P.: Impulsschaltungen. Berlin Springer 1963.
45. Steinbuch K.: Taschenbuch der Nachrichtenverarbeitung. Berlin.
Springer 1962.



9. TARTALOMJEGYZÉK

Bevezetés	3
1. Szekvenciális hálózatok	5
1.1 Bevezetés	5
1.2 Szinkron szekvenciális hálózatok	7
1.3 Aszinkron szekvenciális hálózatok	8
2. Elektronikus tároló áramkörök	11
2.1 Bevezetés	11
2.2 Tároló áramkörök felosztása	11
2.2.1 RS flip-flop	12
2.2.2 Inverz RS flip-flop	15
2.2.3 JK flip-flop	16
2.2.4 Inverz JK flip-flop	18
2.2.5 T flip-flop	18
2.2.6 Inverz T flip-flop	19
2.2.7 RST flip-flop	19
2.2.8 D flip-flop vagy késleltető elem	21
2.2.9 A tárolók logikai egyenleteinek összefoglalása ..	21
2.3 Tároló áramkörök realizálása	22
2.3.1 Bevezetés	22
2.3.2 Statikus vezérlésű aszinkron RS tároló	23
2.3.2.1 RS tároló vezérlési tulajdonságainak vizsgálata	26
2.3.2.2 Aszinkron működésű RS tárolók áramköri megvalósítása	28
2.3.3 Dinamikus vezérlésű tárolók	31
2.3.4 Dinamikus vezérlésű tárolók alkalmazása	33
2.3.4.1 Dinamikus RS flip-flop	33
2.3.4.2 Dinamikus T flip-flop	35
2.3.4.3 Dinamikus RST flip-flop	36
2.3.4.4 Dinamikus JK flip-flop	36
2.4 Gyakorló feladatok	39
3. Szinkron működésű szekvenciális hálózatok regiszterének tervezése	
3.1 Szekvenciális hálózatok felosztása	
3.2 Szinkron szekvenciális hálózatok tervezése	

3.2.1 A tárolók vezérléséhez szükséges bemeneti egyenletek közvetlen meghatározása	48
3.2.2 A tárolók vezérléséhez szükséges bemeneti egyenletek általános meghatározása	54
3.2.2.1 RS flip-flop	55
3.2.2.2 JK flip-flop	57
3.2.2.3 T flip-flop	58
3.2.2.4 RST flip-flop	60
3.2.2.5 D flip-flop	61
3.2.2.6 Összefoglalás	62
3.3 Gyakorló feladatok	63
4. Szekvenciális funkcionális egységek	64
4.1 Szekvenciális generátorok	64
4.1.1 Gyakorló feladatok	66
4.2 Számláncok	67
4.2.1 Bináris számláncok	67
a) Soros számláncok	67
b) Párhuzamos számláncok	67
c) Reverzibilis számláncok	72
4.2.2 Decimális számláncok	75
a) Gyűrűs számláncok	75
b) Párhuzamos beírású számláncok	79
4.2.3 Gyakorló feladatok	83
4.3 Szekvenciális átalakítók	85
4.3.1 Soros üzemi Gray/bináris konverzió	86
4.3.2 Kettes komplementképző	89
4.3.3 Lineáris szekvenciális hálózatok	92
4.3.4 Gyakorló feladatok	96
4.4 Üzem mód átalakítók	100
4.5 Soros működésű bináris összeadó, kivonó egység	102
4.5.1 Soros működésű összeadó egység	102
4.5.2 Soros működésű kivonó egység	109
4.5.3 Összeadó-kivonó egység	112
4.5.4 Gyakorló feladatok	113
4.5.4.1 Aritmetikai egységek tervezése	113
4.5.4.2 Gyakorló feladatok	117
5. Szinkron szekvenciális hálózatok	121
5.1 Szekvenciális hálózatok analízise	121
5.2 Állapot táblázat egyszerűsítése HUFFMAN-MEALY módszerrel	123
5.3 Állapot táblázat egyszerűsítése lépcsős táblázattal	128
5.4 Szekvenciális hálózatok tervezése	132
5.4.1 Egyszerű hibajelző	132
5.4.2 Ciklikus hibajelző	139
5.4.3 Hibajelző BCD kód vizsgálatára	143

5.5 Gyakorló feladatok	149
6. Függelék	154
6.1 Minterm tábla	154
6.2 Maxterm tábla	154
6.3 Logikai függvények realizálási módjai	158
7. Jelmagyarázat	165
8. Irodalom	167
9. Tartalomjegyzék	171

J 5-872