



# 5. Hét

## Sorrendi hálózatok

Digitális technika  
2015/2016





# Bevezető példák

Ó  
B  
U  
D  
A  
I  
  
E  
G  
Y  
E  
T  
E  
M





# Példa 1: Italautomata

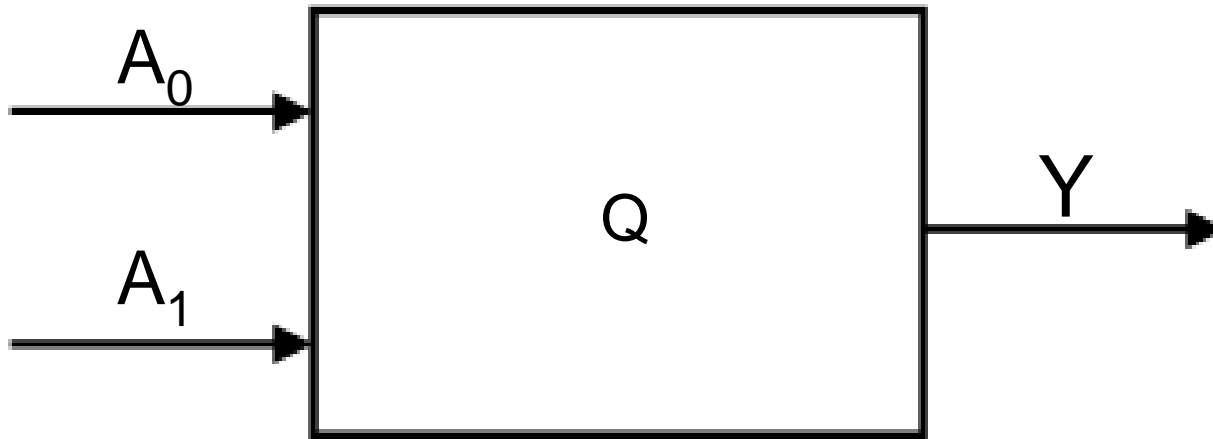
Legyen az általunk vizsgált rendszer egy italautomata, amelyről az alábbi dolgokat tudjuk:

- ❖ 150 Ft egy üdítő
- ❖ 50 és 100 Ft-os érmét fogad el,
- ❖ é visszaad





# Italautomata





# Italautomata

## ❖ Bemenetek:

❖ 100Ft;

❖ 50 Ft

## ❖ Kimenetek;

❖ üdítő ki;

❖ 50 Ft ki;

## ❖ Belső állapotok:

❖ Start állapot:

❖ Bedobtunk 100 Ft-ot : ,

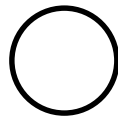
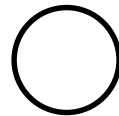
❖ Bedobtunk 50 Ft-ot:



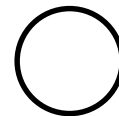


# Italautomata

START állapot



50 Ft bedobva

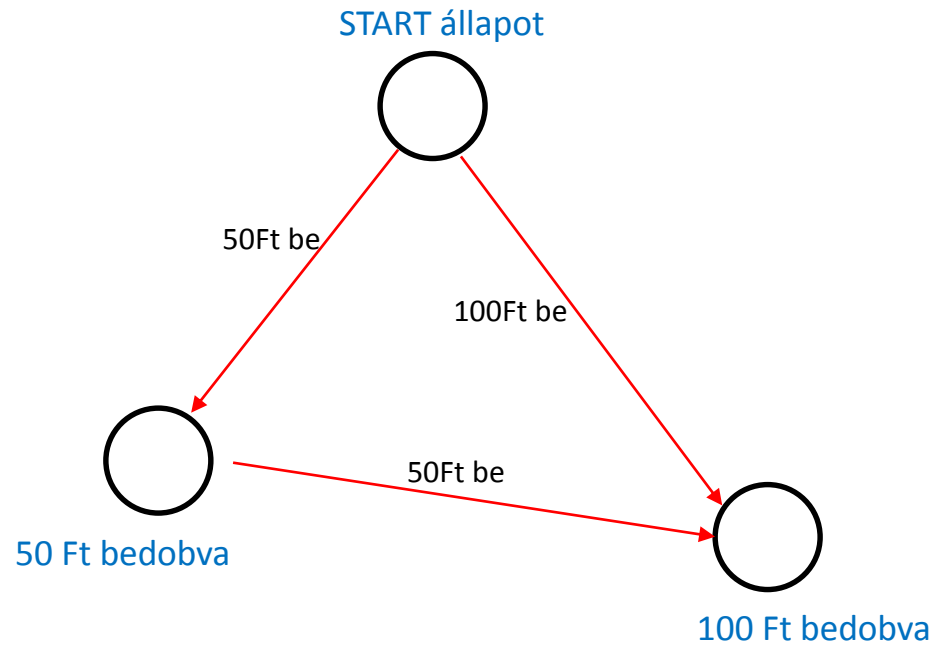


100 Ft bedobva



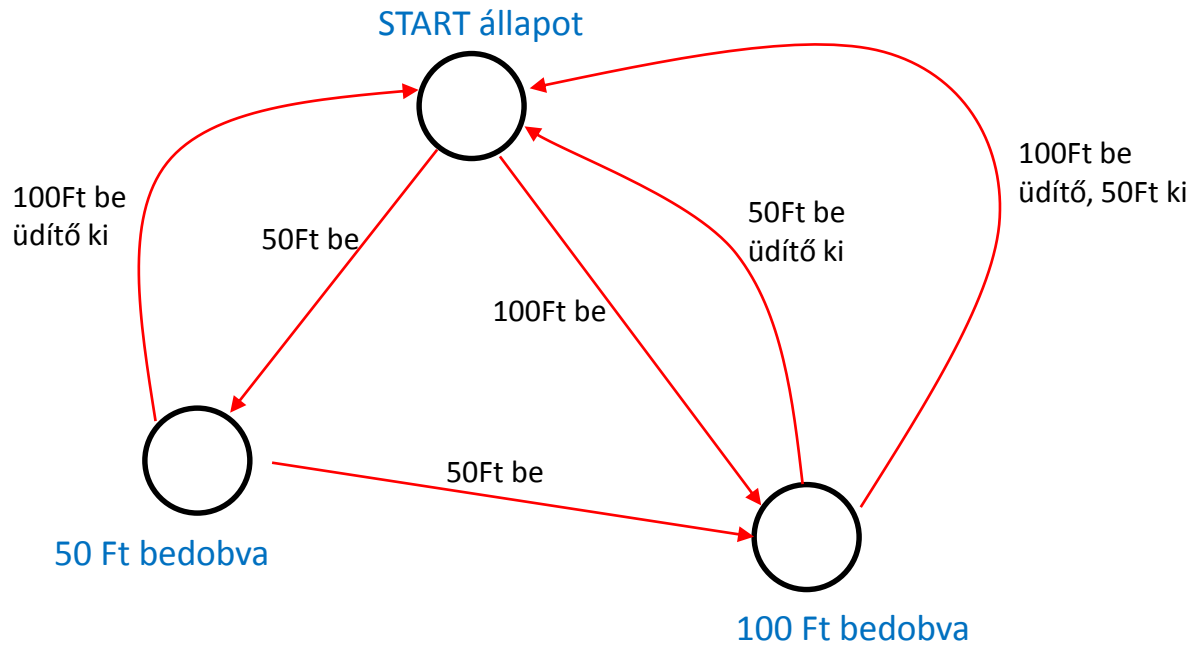


# Italautomata





# Italautomata







# Italautomata

## ❖ Bemenetek:

❖  $A_1$ : 100Ft;

❖  $A_0$ : 50 Ft

## ❖ Kimenetek;

❖  $Y_1$ : üdítő ki;

❖  $Y_0$ : 50 Ft ki;

## ❖ Belső állapotok:

❖ Start állapot: 00,

❖ Bedobtunk 100 Ft-ot : 01,

❖ Bedobtunk 50 Ft-ot: 10.





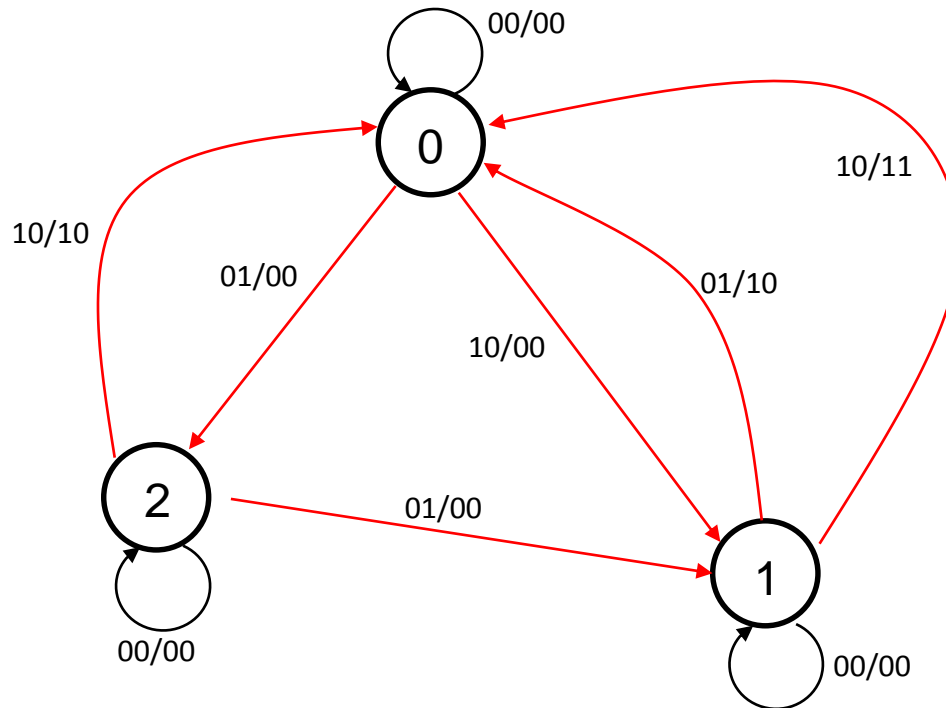
# Italautomata - táblázat

Előző állapot	Bemenet ( $x_1x_0$ ) 100/50Ft			
	00	01	10	11
0 (0 Ft)	0	2	1	x
1 (100 Ft)	1	0	0	x
2 (50 Ft)	2	1	0	x



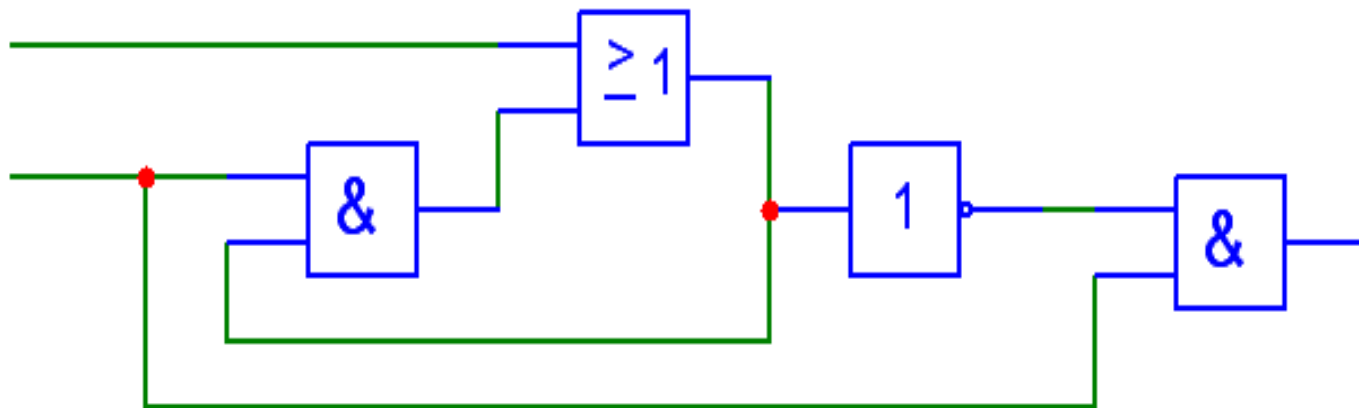


# Italautomata



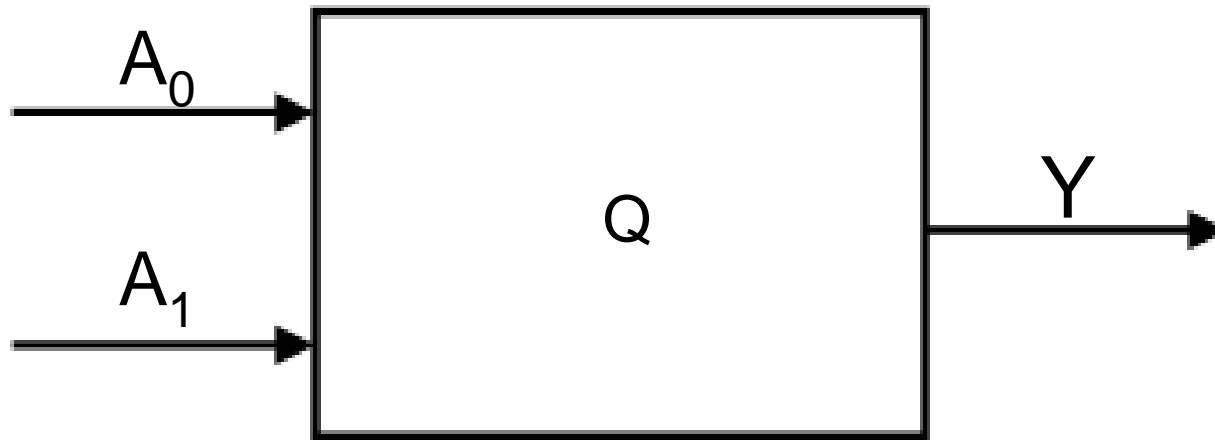


# Példa 2: Hálózat



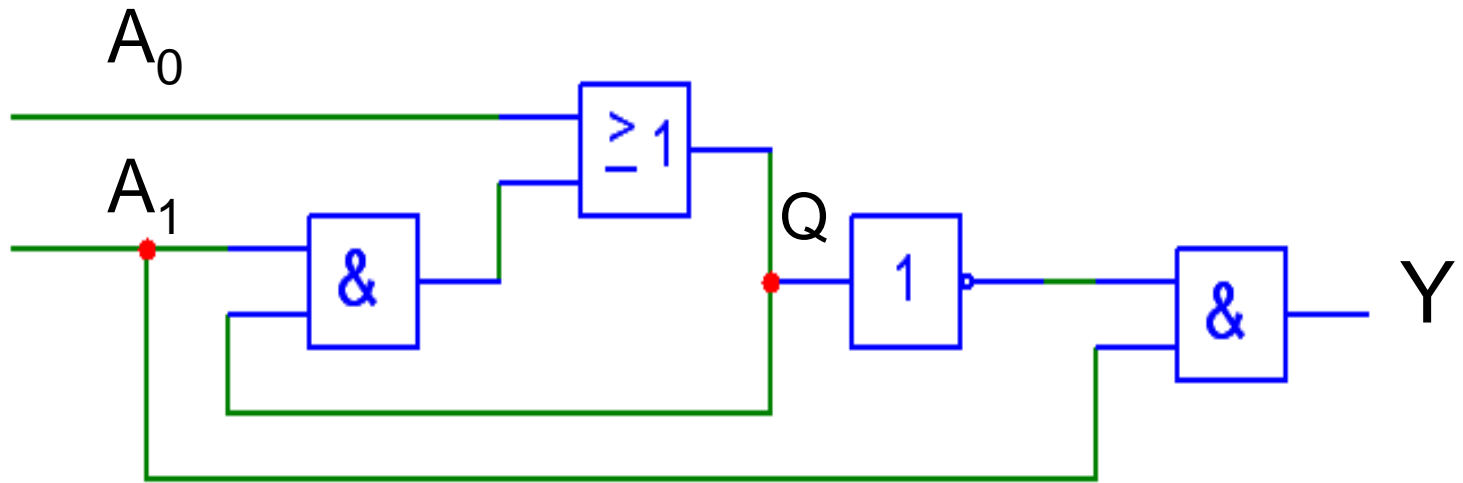


# Hálózat



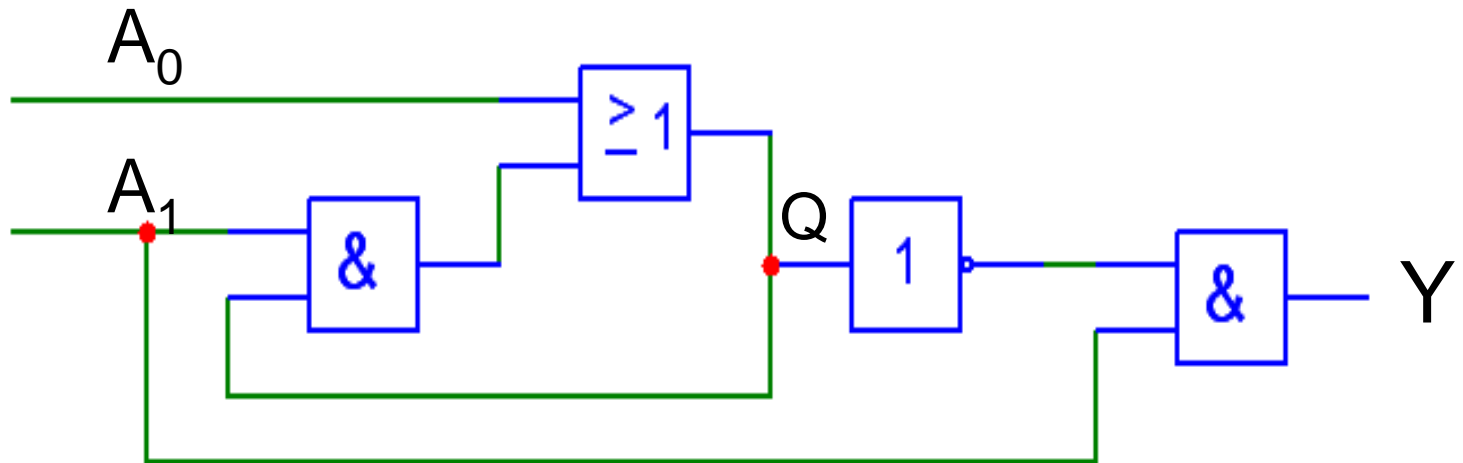


# Hálózat





# Hálózat



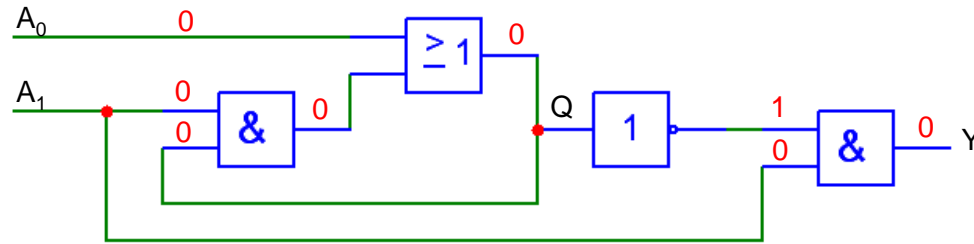
$$Q_{n+1} = A_0 + A_1 Q$$

$$Y = A_1 \cdot \bar{Q}$$





# Hálózat



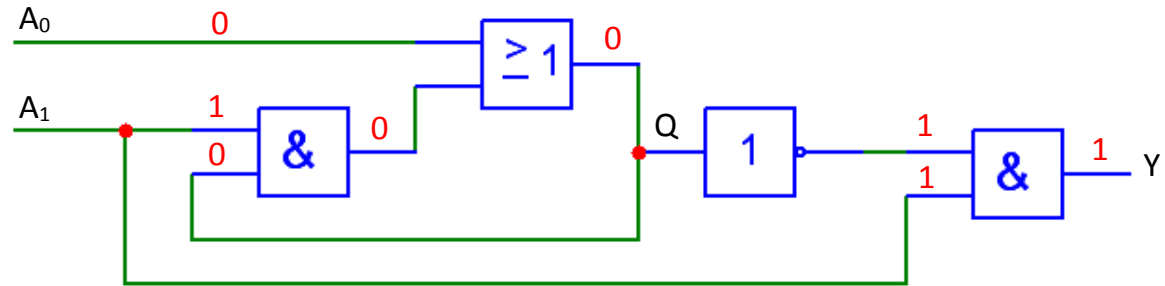
$A_0$	$A_1$	$Q$	$Y$
0	0	0	0







# Hálózat

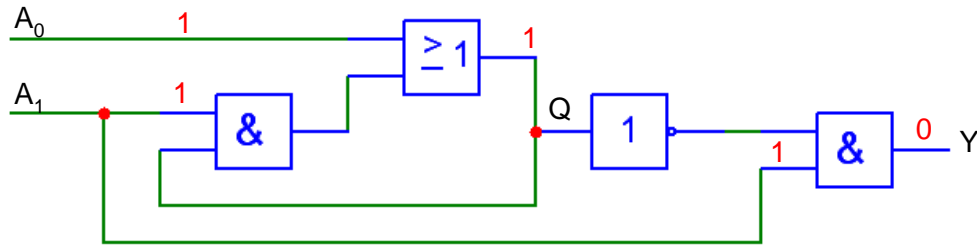


$A_0$	$A_1$	$Q$	$Y$
0	0	0	0
0	1	0	1





# Hálózat

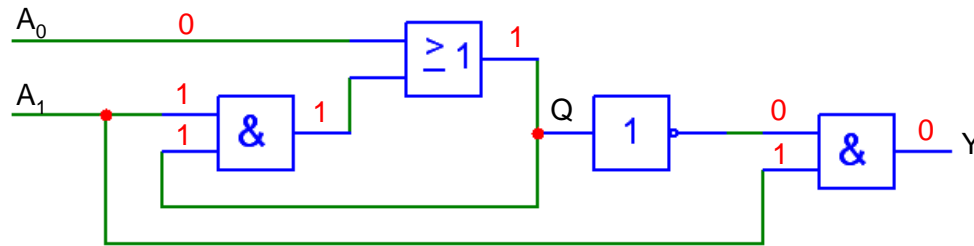


$A_0$	$A_1$	$Q$	$Y$
0	0	0	0
0	1	0	1
1	1	0	0





# Hálózat

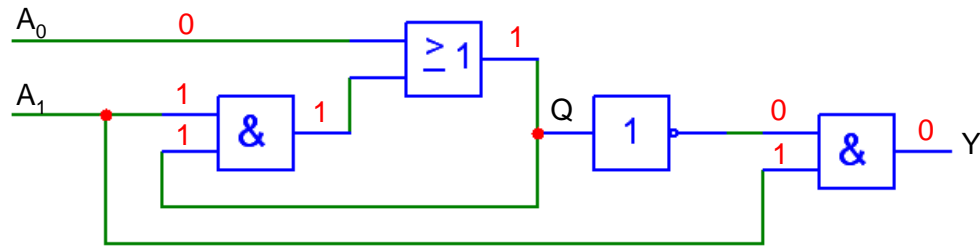


$A_0$	$A_1$	$Q$	$Y$
0	0	0	0
0	1	0	1
1	1	0	0
0	1	0	0





# Hálózat



$A_0$	$A_1$	Q	Y
0	0	0	0
0	1	0	1
1	1	0	0
0	1	0	0





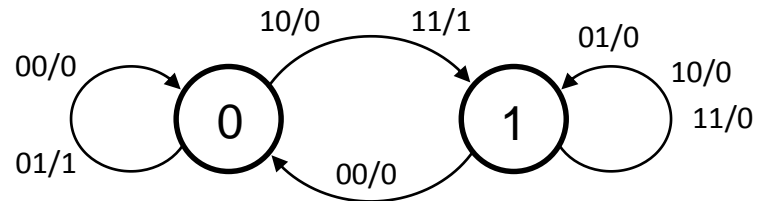
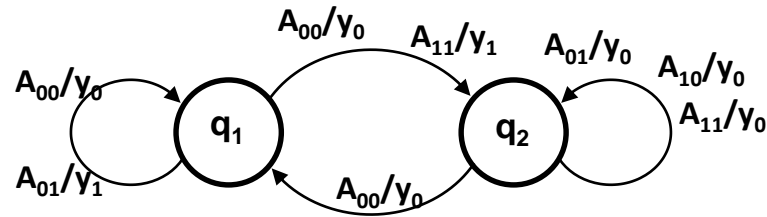
# Hálózat - Tábla

Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0





# Hálózat





# Logikai hálózat

**Logikai hálózatnak** nevezzük azokat a rendszereket, melyeknek bemeneti illetve kimeneti jelei **logikai jelek**, a kimeneti jeleket a bemeneti jelek függvényében többé-kevésbé bonyolult logikai műveletsorozat eredményeként állítják elő.





# Logikai hálózat típusai

- ❖ Kombinációs hálózat
- ❖ Sorrendi hálózat







# Kombinációs hálózat

- ❖ Kimeneti jelei csak a bemeneti jelek pillanatnyi értékétől függnnek
- ❖ „Emlékezet” nélküli hálózatok.



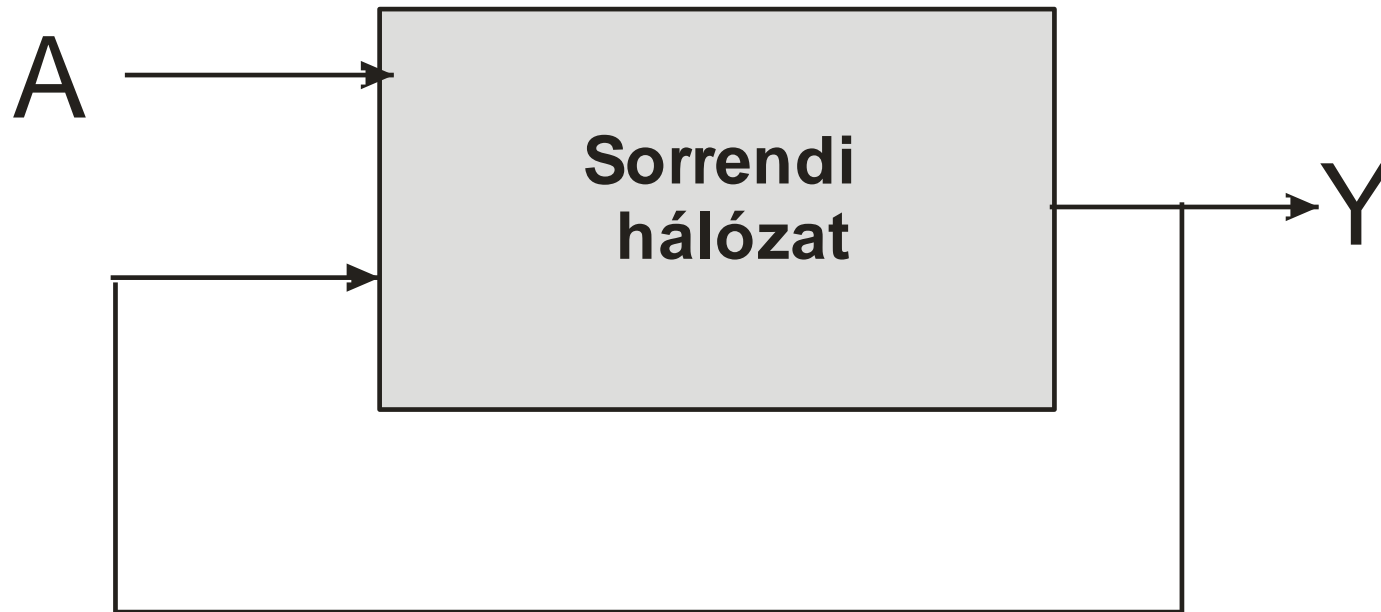


# Kombinációs hálózat



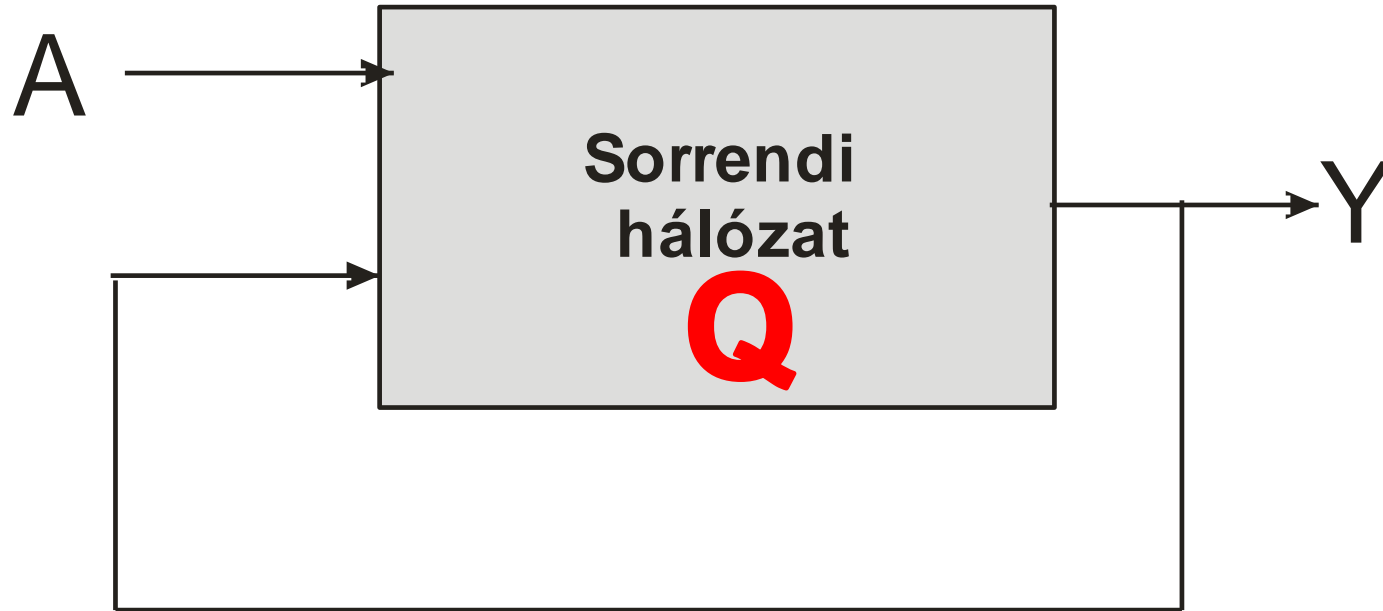


# Sorrendi hálózat





# Sorrendi hálózat



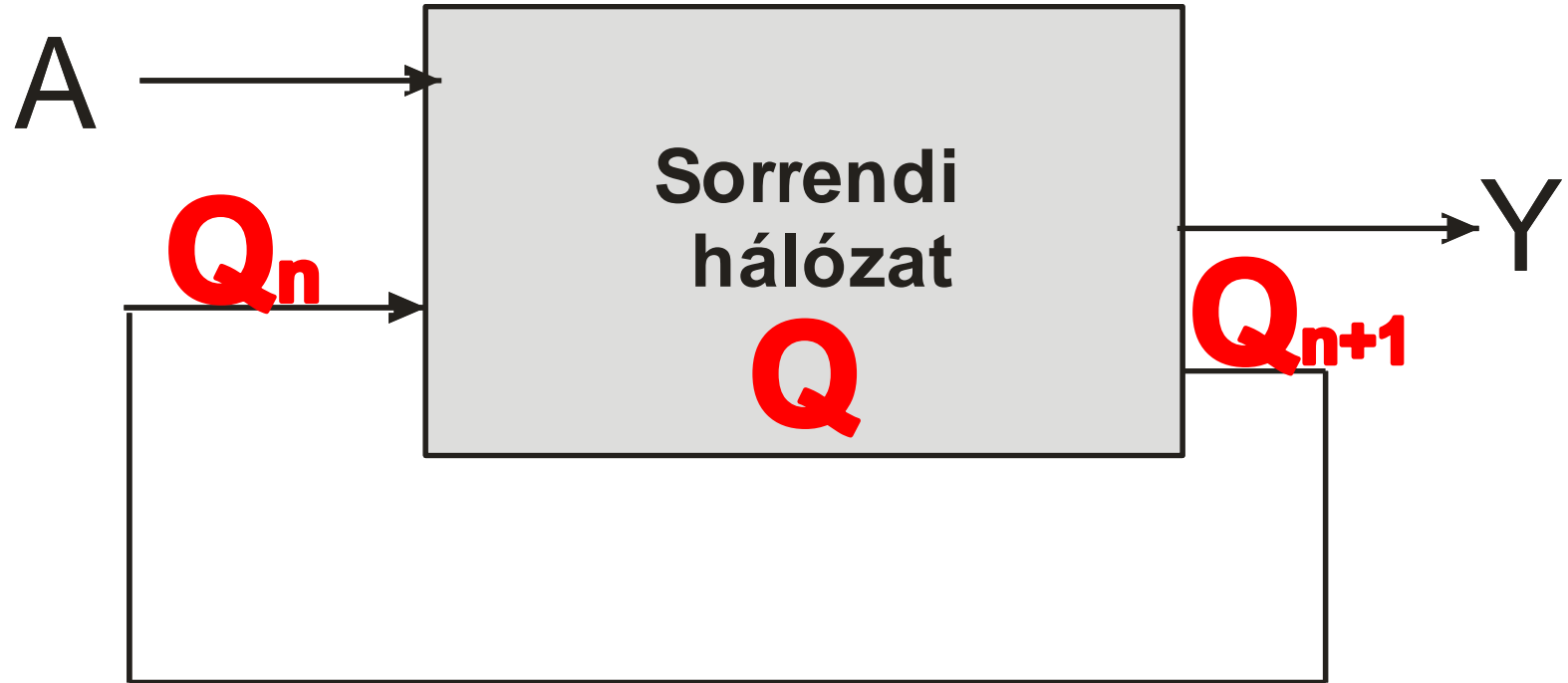
Belső állapot





# Sorrendi hálózat

Primer változó



Szekunder változó





# A sorrendi hálózat működése

- ❖ Egy  $n$  hosszúságú bemeneti sorozat (szekvencia)
- ❖  $n$  hosszúságú belső állapot (szekunder változó) szekvencia jön létre,
- ❖  $n$  hosszúságú kimeneti szekvencia generálódik.
- ❖ Ha  $n$  véges: **véges sorrendi automatáról** beszélünk.





# A sorrendi hálózatok leírása

- ❖ Függvény
- ❖ Modell
- ❖ Állapottábla
- ❖ Állapotgráf





# Függvény

$$f_y = (A, Q) \Rightarrow Y \quad f_{Q^{n+1}} = (A, Q) \Rightarrow Q^{n+1}$$







# A sorrendi hálózat típusai

- ❖ Szinkron hálózat
- ❖ Aszinkron hálózat





# Aszinkron hálózat

$$f'_y = (Q) \Rightarrow Y$$

$$f_{Q^{n+1}} = (A, Q) \Rightarrow Q^{n+1}$$

Stabil állapot :  $Q = Q^{n+1}$

Ha ez nem igaz: **instabil**

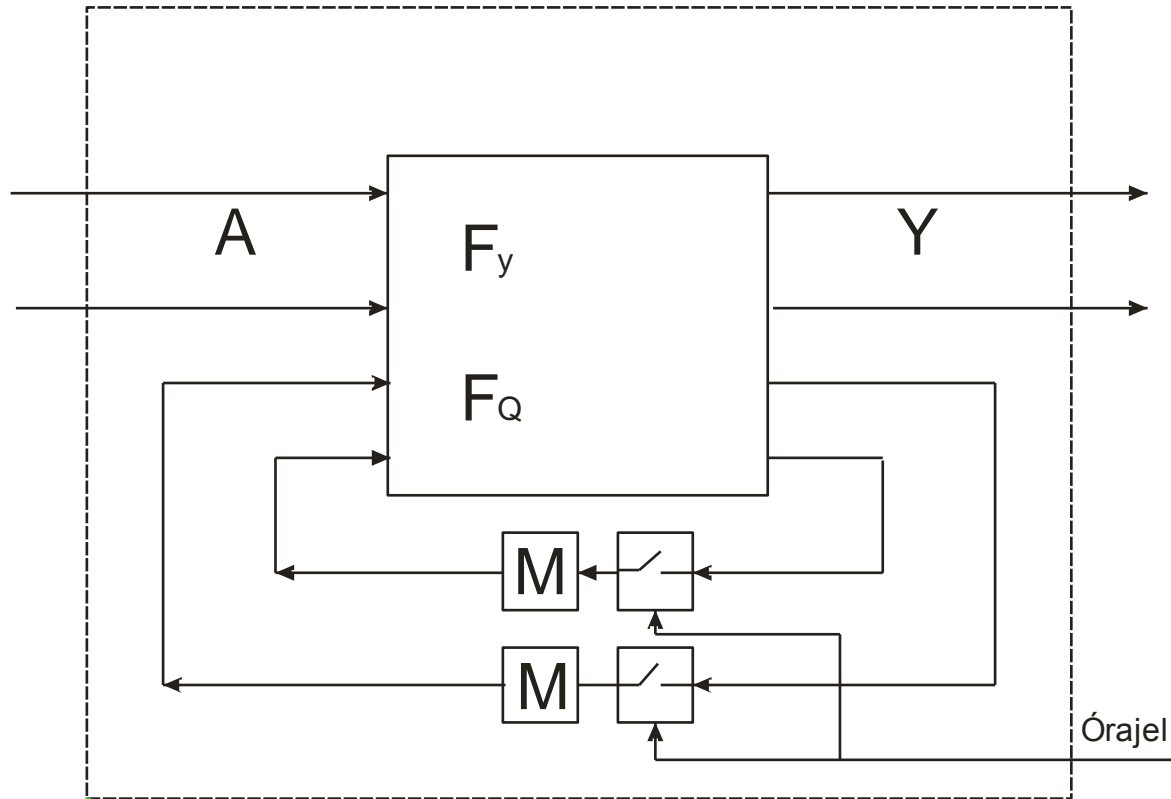
**Oszcilláció:** akkor lép fel ha nem alakul ki stabil állapot, hanem csak instabil

A csak stabil állapotban változhat!





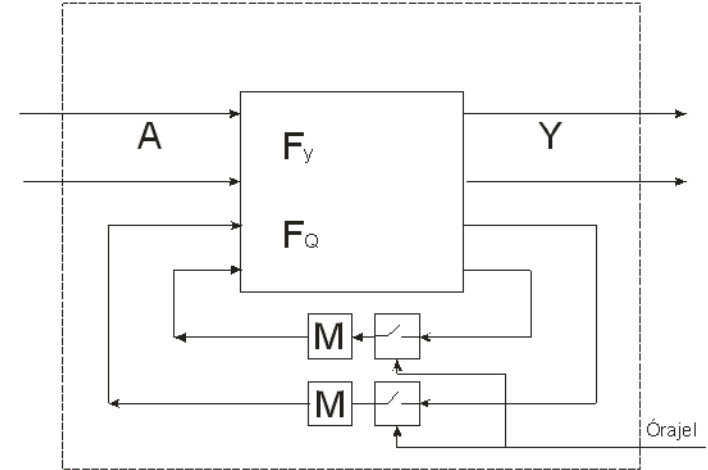
# Aszinkron hálózat





# Aszinkron hálózat

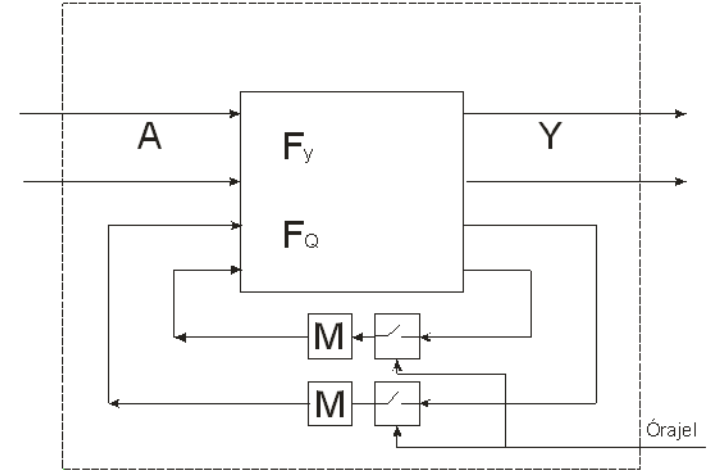
- ❖ Kapcsolók: periodikus (négyszögjellel)
- ❖ M:
  - ❖ Azt adja tovább, ami a zárt állapotban érkezett
  - ❖ Memória: megőrzi az utolsó értéket nyitott állapotban
- ❖ Az órajel logikai 1 szintjének időtartama alatt a visszacsatoló ágak zártak, 0 esetén pedig nyitottak.





# Aszinkron hálózat

- ❖ az instabil állapotok miatt a szükséges
- ❖ szekunder változók száma általában nagyobb, mint szinkron esetben
- ❖ többek között ezért szinkron tervezésük **bonyolultabb.**
- ❖ a bemeneti változások gyakoriságát, vagyis a működési sebességet csak az **építőelemek működési sebessége** és a **jelterjedési késleltetések** okozzák.
- ❖ A csak stabil állapotban változhat!!





# Szinkron hálózat

$$f'_y = (A, Q) \Rightarrow Y$$

$$f_{Q^{n+1}} = (A, Q) \Rightarrow Q^{n+1}$$

Stabil állapot :  $Q = Q^{n+1}$

Ha ez nem igaz: **instabil**

**Oszcilláció:** akkor lép fel ha nem alakul ki stabil állapot, hanem csak instabil

DE itt a stabil/ instabil állapot nem fontos: A változhat instabil állapotban is





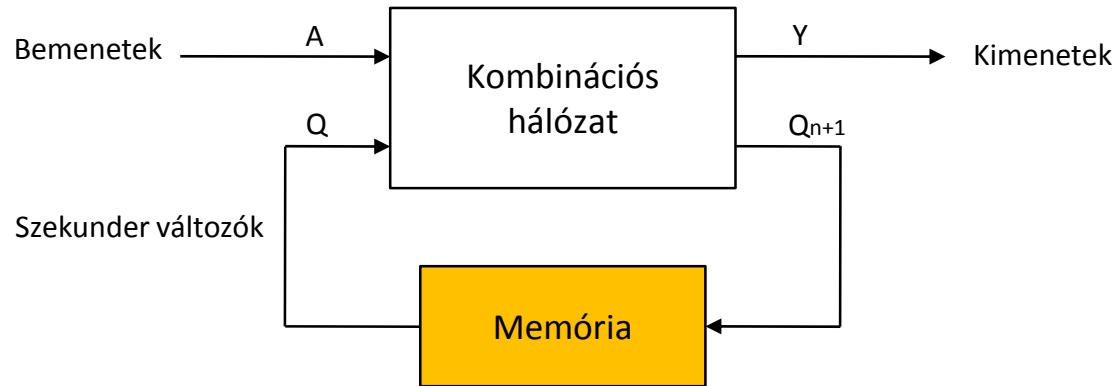
# Szinkron hálózat

- ❖ a stabil és az instabil állapotok **nincsenek értelmezve**,
- ❖ a szükséges **szekunder változók** száma ezért általában **kisebb**, mint aszinkron esetben,
- ❖ ezért logikai tervezésük **egyszerűbb**.
- ❖ a **működési sebességet az órajel frekvenciája** korlátozza
- ❖ ezért általában **lassúbbak**, mint az ütemezés nélküli szinkron hálózatok
- ❖ a bemeneti változásokra és a kimeneti kombináció értelmezésére **szinkronizációs feltételeknek** kell teljesülniük
- ❖ logikai tervezésük után, megvalósításuk során biztosítani **kell** a szinkronizációs feltételeket.





# A sorrendi hálózat modellje







# Modellek

- ❖ Mealy – modell
- ❖ Moore - modell

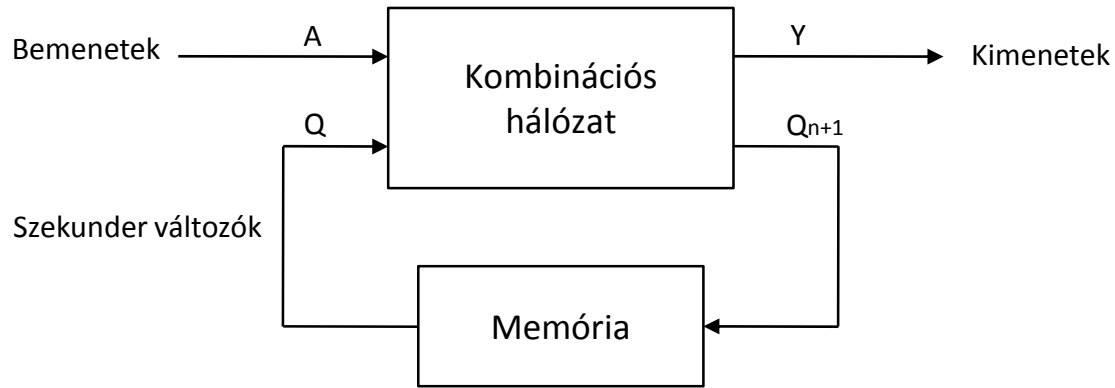




# Mealy - modell

$$f_y = (A, Q) \Rightarrow Y$$

$$f_{Q^{n+1}} = (A, Q) \Rightarrow Q^{n+1}$$

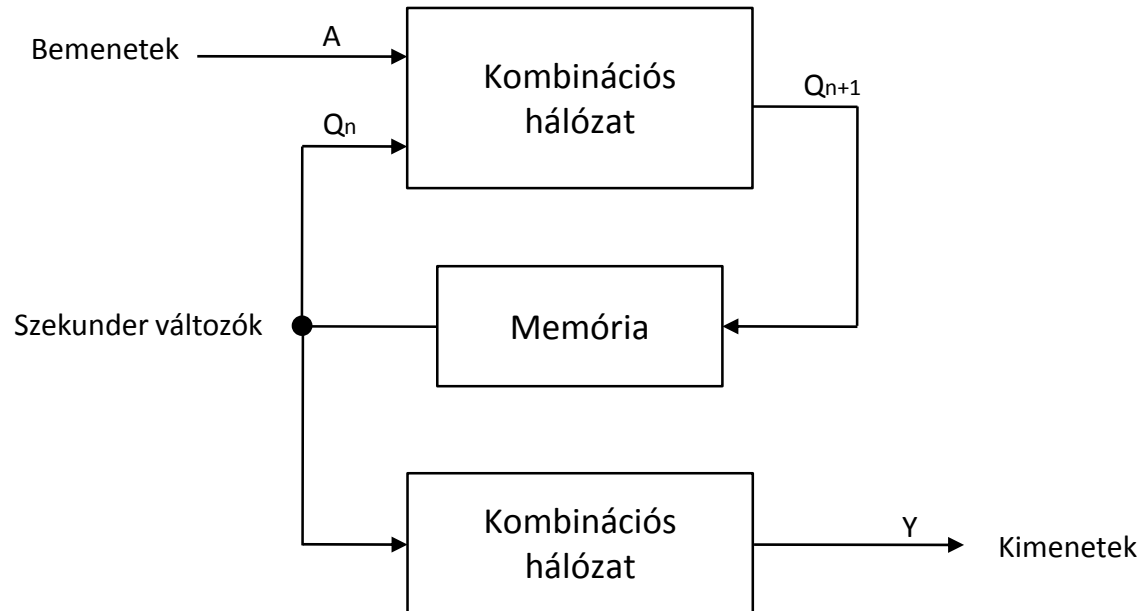




# Moore - modell

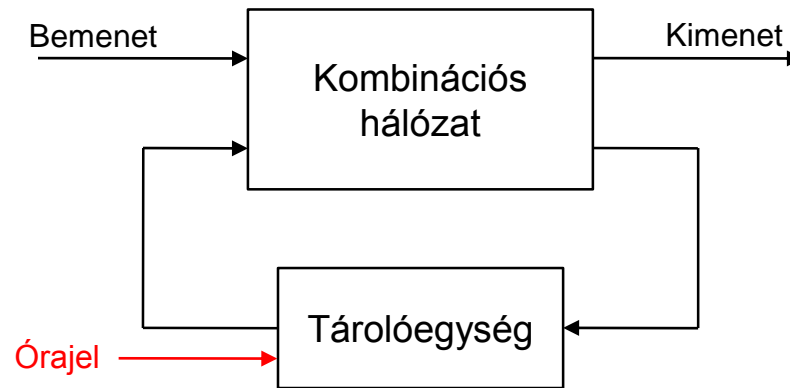
$$f_y = (Q) \Rightarrow Y$$

$$f_{Q^{n+1}} = (A, Q) \Rightarrow Q^{n+1}$$





# Szinkron hálózatok

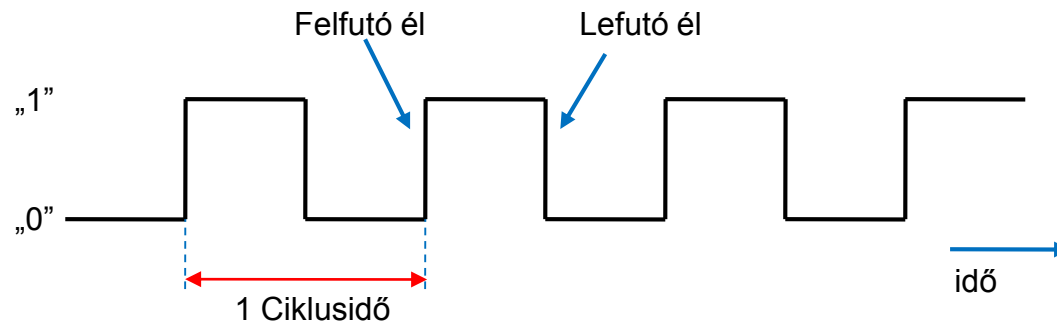


- ❖ Két fő eleme
  - ❖ Tárolóegység (Memória)
    - ❖ A korábbi bemeneti kombinációkra vonatkozó információ tárolására
  - ❖ Bemeneti kombinációs hálózat
    - ❖ A kimeneti jel előállítása
    - ❖ A tárolandó információ előállítása
      - ❖ A bemeneti kombinációkból és az előzőleg eltárolt információk együtt határozzák meg a következő ciklusban eltárolandó információt
- ❖ Fontos különbség az aszinkron sorrendi hálózatokhoz képest
  - ❖ A jelváltozások nem futnak rögtön végig a hálózaton, csak a következő ciklusban hatnak





# Szinkronizáció



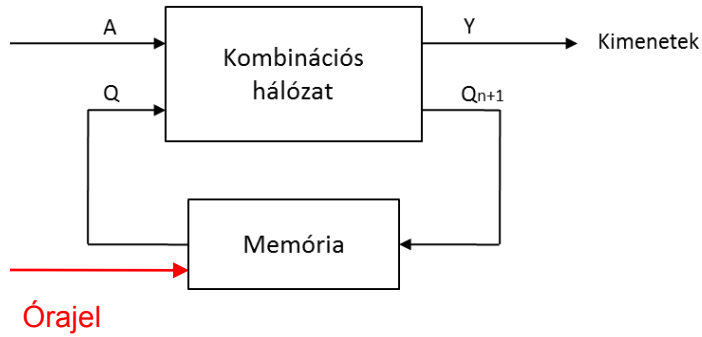
- ❖ Minden változás az órajellel időzítve, azzal szinkronizálva megy végbe, előre pontosan definiált időpillanatban, az órajel fel- vagy lefutó élének megérkezését követően



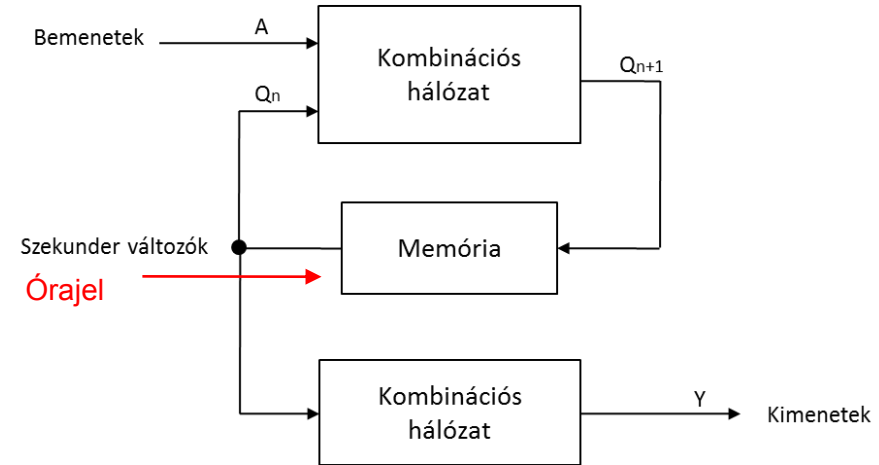


# Modell

Mealy modell



Moore modell







# Állapottábla

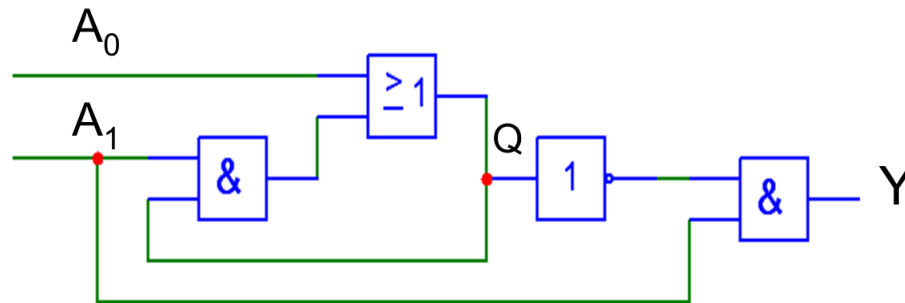
- ❖ Bemeneti változók
  - ❖ Bemenetek
  - ❖ Szekunder változók
- ❖ Kimeneti változók
  - ❖ Kimenetek
  - ❖ Szekunder változók







# Állapottábla



Q	$A_0$	$A_1$	$Q_{n+1}$	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0





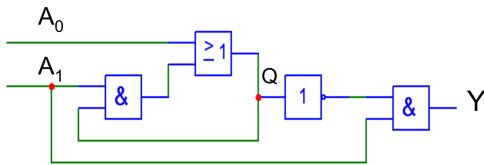
# Italautomata - táblázat

Előző állapot	Bemenet ( $x_1x_0$ ) 100/50Ft			
	00	01	10	11
0 (0 Ft)	0	2	1	x
1 (100 Ft)	1	0	0	x
2 (50 Ft)	2	1	0	x





# Hálózat - Tábla



Q	$A_0$	$A_1$	$Q_{n+1}$	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0





# Állapotgráf

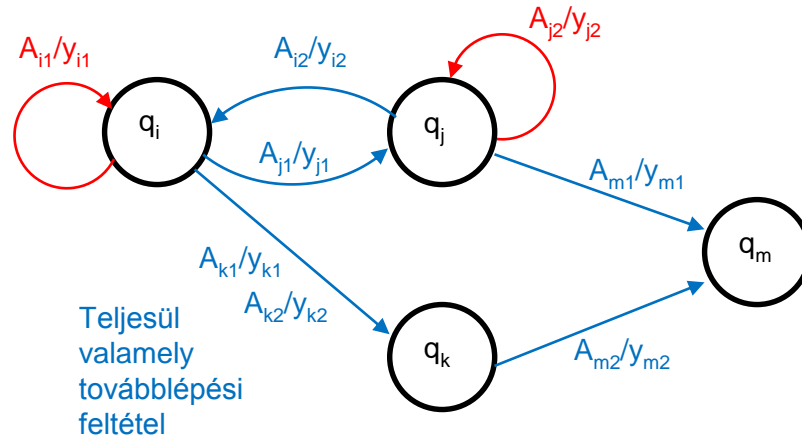
- ❖ Grafikus
- ❖ belső állapotok - csomópontok
- ❖ A csomópontokat összekötő irányított élek (nyilak) az egyik állapotból a másikba történő átmenetet reprezentálják.
- ❖ Az éleken az átmenetet előidéző bemeneti **A** kombináció szerepel.
- ❖ Emellett az **Y** kimeneti értékeket is gyakran fel szokás tüntetni





# Állapotgráf

Egyik  
továblépési  
feltétel sem  
teljesül



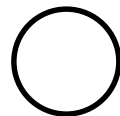
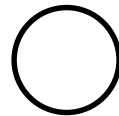
Teljesül  
valamely  
továblépési  
feltétel



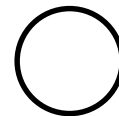


# Italautomata

START állapot



50 Ft bedobva

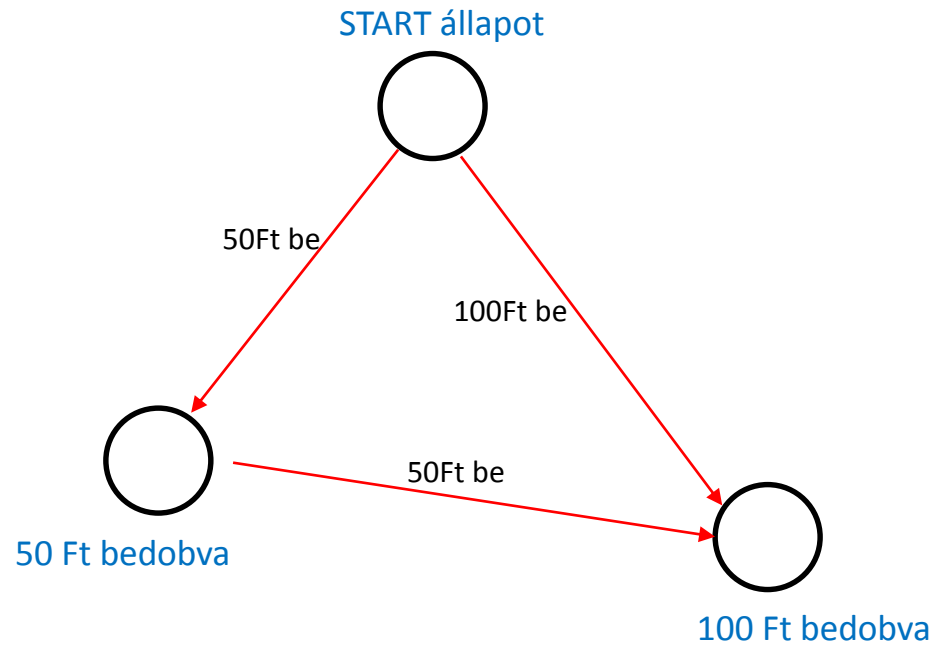


100 Ft bedobva



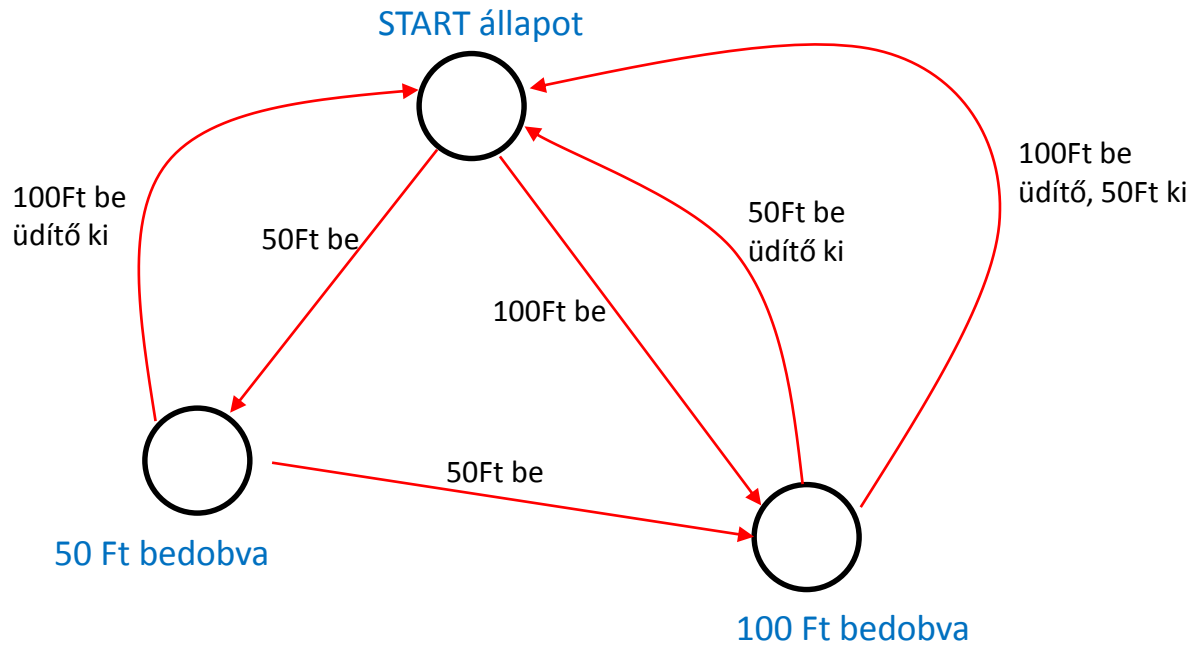


# Italautomata





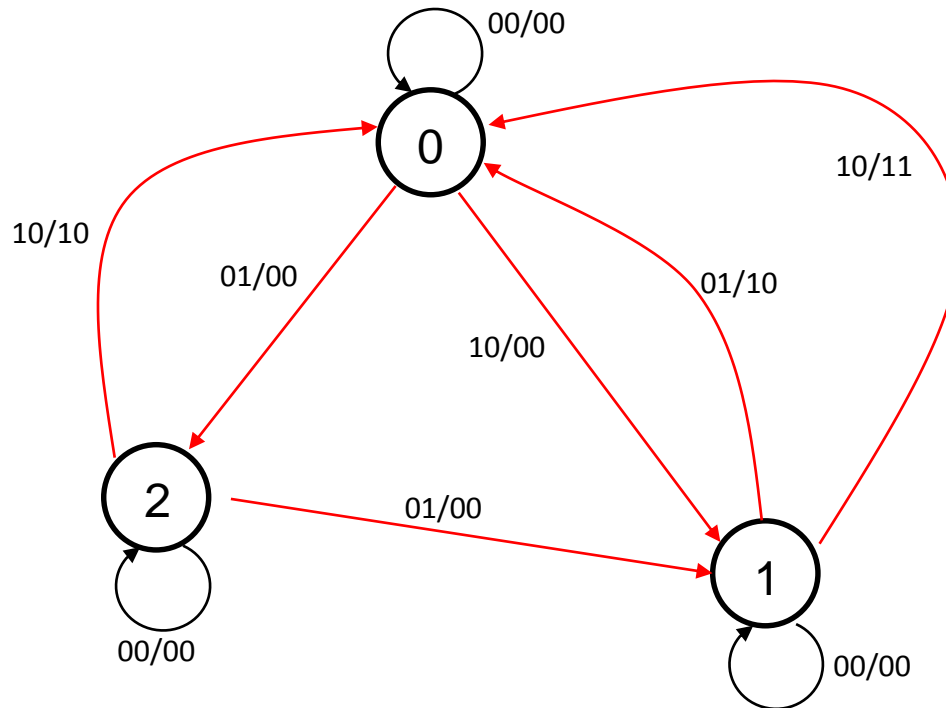
# Italautomata





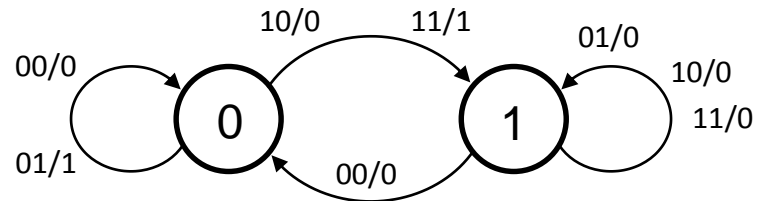
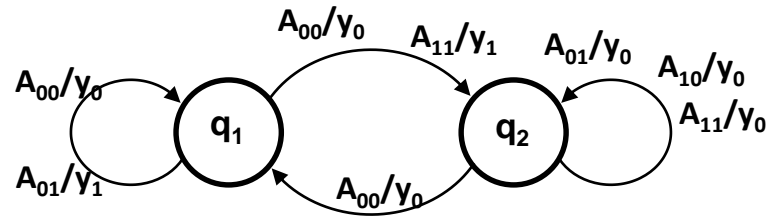


# Italautomata





# Hálózat





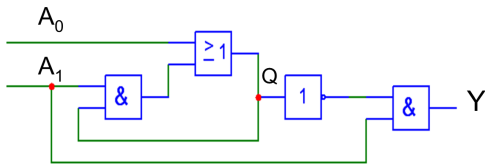
# Vezérlési táblázat

- ❖ A vezérlési táblázat az állapotábra célszerűen átalakított formája,
- ❖ az oszlopok a bemenő jelek,
- ❖ a sorok pedig a késleltetés után előállt visszacsatolt jelek.





# Vezérlési tábla előállítás

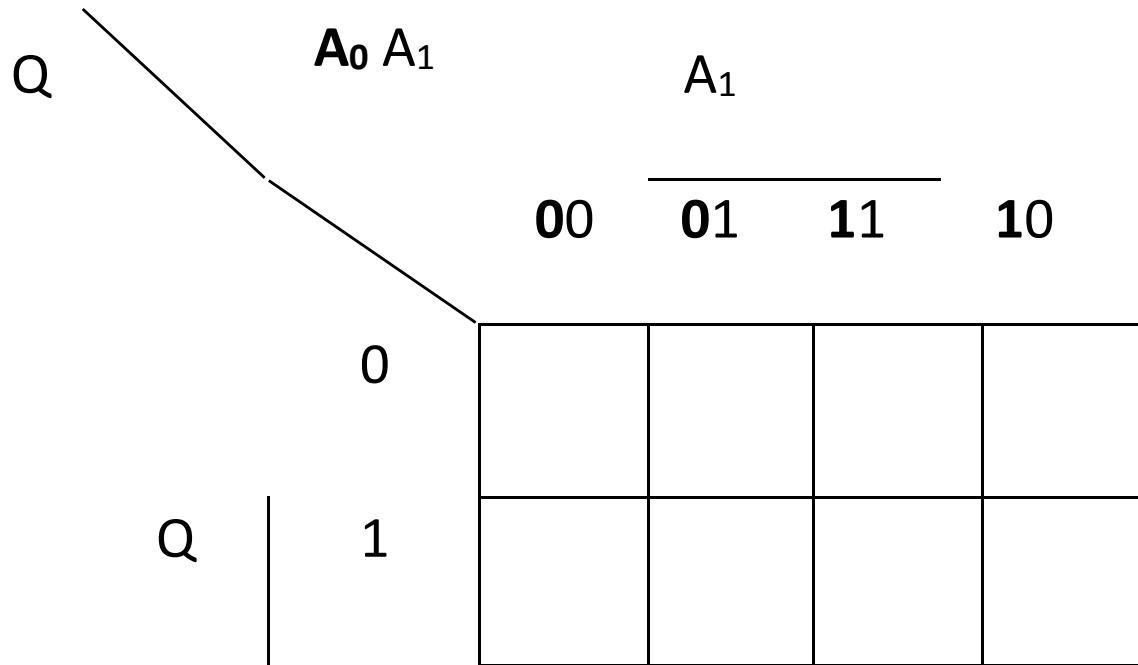


i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0



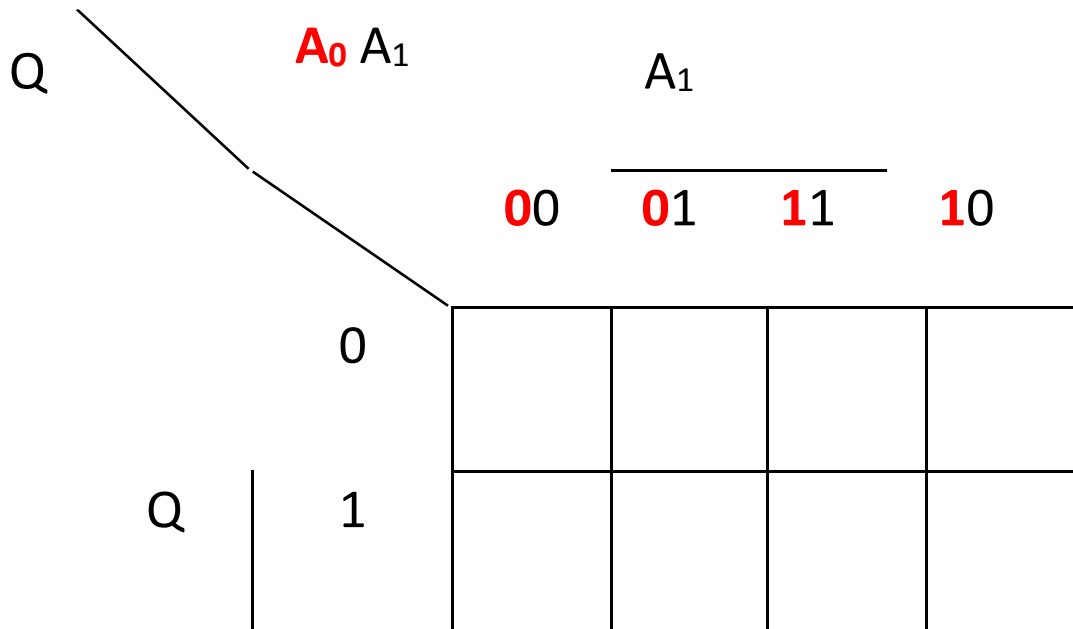


# Vezérlési tábla





# Vezérlési tábla





# Vezérlési tábla

$A_0 A_1$

$A_1$

00

01

11

10

0

$\overline{Q}A_0A_1$

$\overline{Q}A_0A_1$

$\overline{Q}A_0A_1$

$\overline{Q}A_0\overline{A}_1$

1

$Q\overline{A}_1A_2$

$Q\overline{A}_0A_1$

$QA_0A_1$

$QA_0\overline{A}_1$

Q





# Vezérlési tábla

Q

**A<sub>0</sub>** A<sub>1</sub>

A<sub>1</sub>

**00**

**01**

**11**

**10**

0

000

001

011

010

Q

1

100

101

111

110

Ó  
B  
U  
D  
A  
I  
E  
G  
Y  
E  
T  
E  
M







# Vezérlési tábla

Q

**A<sub>0</sub>** A<sub>1</sub>

A<sub>1</sub>

00    01    11    10

0	0	1	3	2
1	4	5	7	6

Q

Q

Detailed description: A control table for a 2-bit counter. The vertical axis is labeled 'Q' and has values 0 and 1. The horizontal axis is labeled 'A<sub>0</sub> A<sub>1</sub>' and has values 00, 01, 11, and 10. The table contains the next state values for each combination of current state and inputs. A diagonal line labeled 'Q' starts from the top left and points to the first row of the table. A vertical line labeled 'Q' is positioned to the left of the table.





# Vezérlési tábla

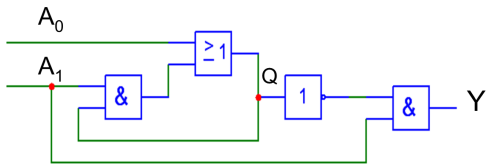
Az állapottáblából kiválasztjuk azokat a sorokat, ahol  $Q$  értéke 1 lesz a folyamat során.

- ❖ Ezt vezetjük be a vezérlési táblába
- ❖ Meghatározzuk a stabil állapotokat
- ❖ Kiolvassuk a vezérlés folyamatát





# Vezérlési tábla előállítás



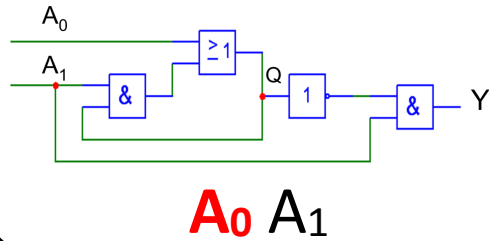
i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0

$$Q_{n+1} = 1$$





# Vezérlési tábla előállítás



$$Q_{n+1} = 1$$

		$A_1$			
		$00$	$01$	$11$	$10$
$Q$	$0$	0	0	1	1
	$1$	0	1	1	1





# Vezérlési tábla előállítás

- ❖ A stabil állapotokat az fogja jelenteni, ahol

$$Q_n = Q_{n+1}$$

- ❖ tehát adott esetben 0, 1, 5, 6, 7 értékeknél.

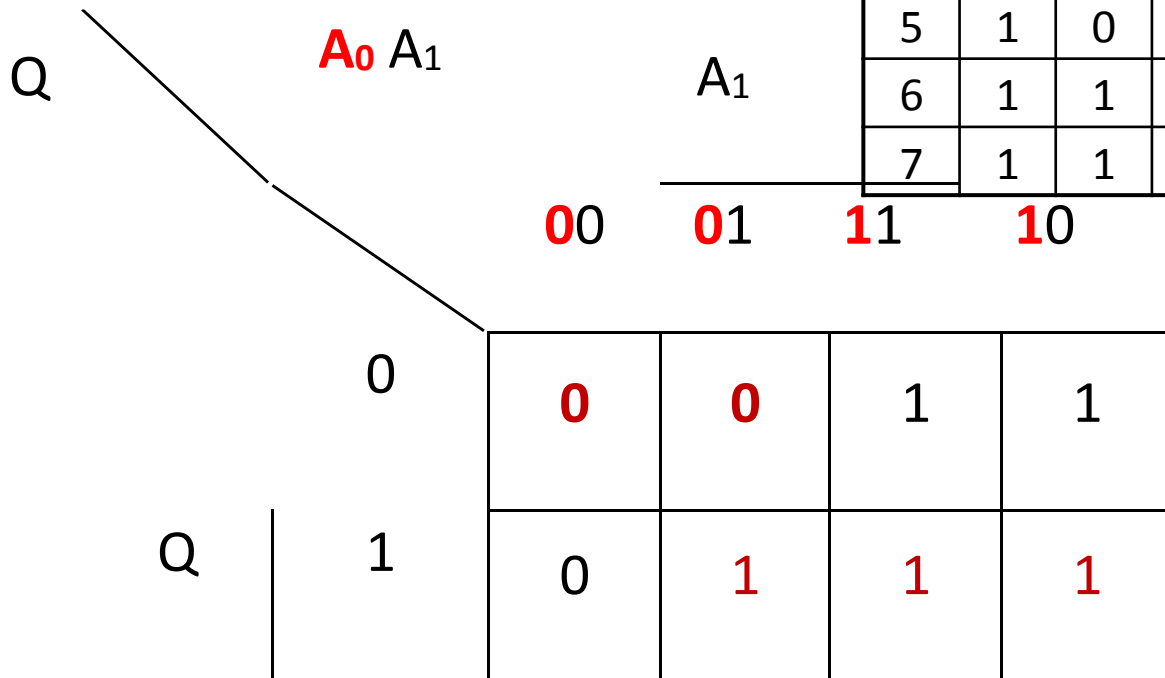
i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0





# Vezérlő tábla kiolvasása

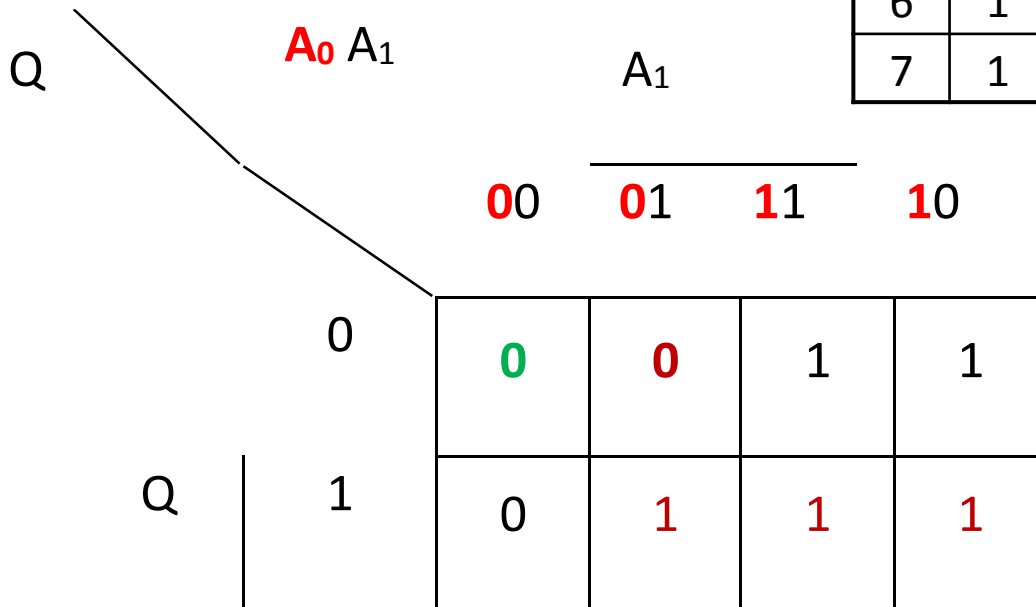
i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0





# Vezérlő tábla kiolvasása

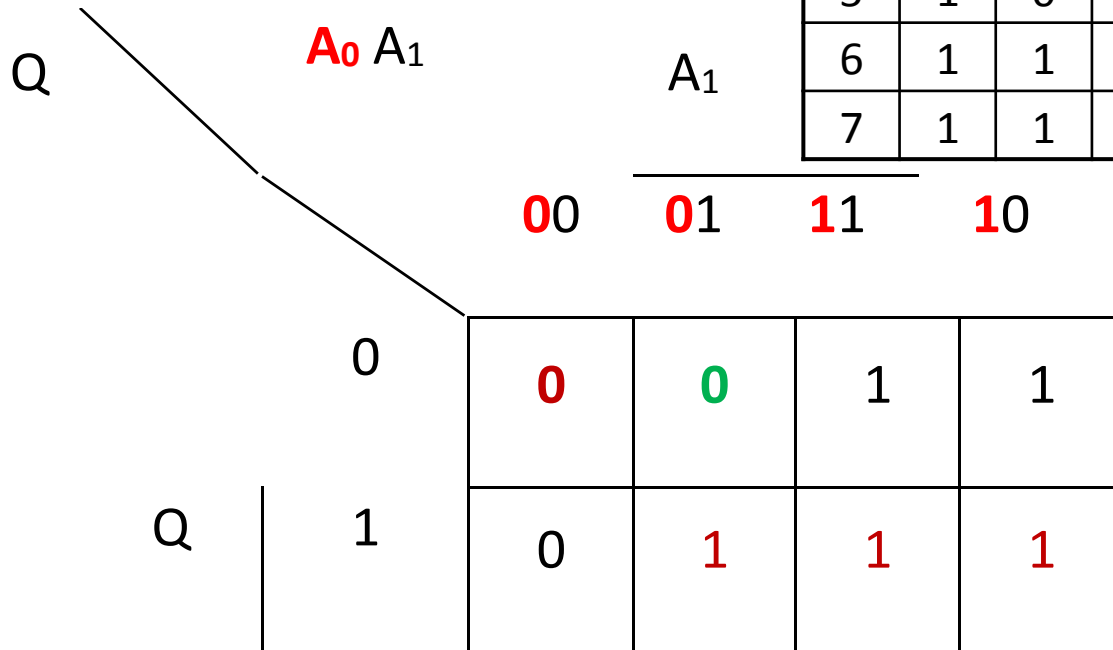
i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0





# Vezérlő tábla kiolvasása

i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0

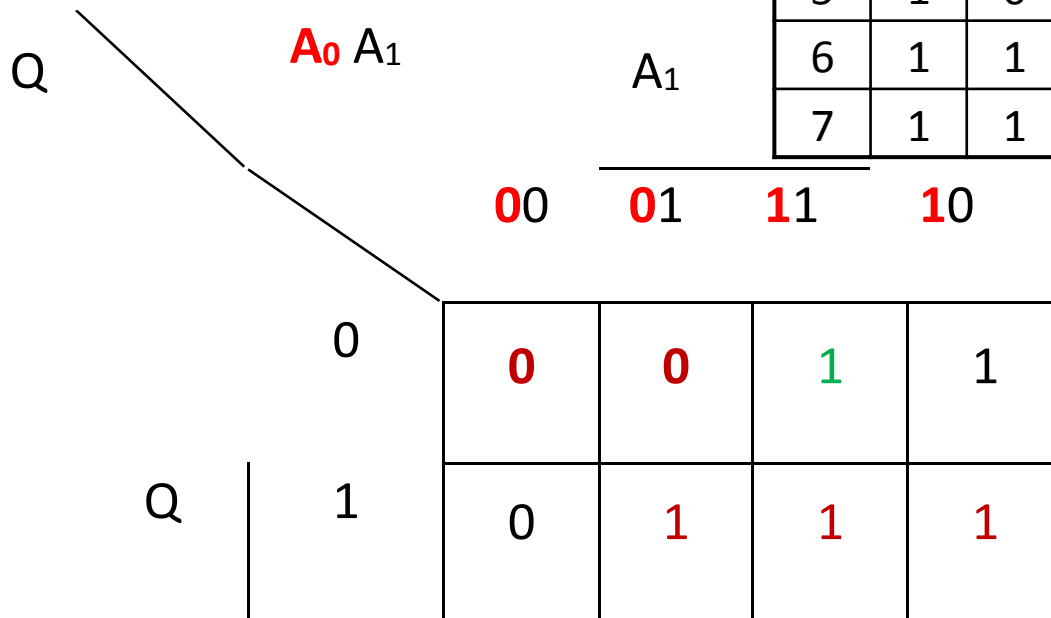






# Vezérlő tábla kiolvasása

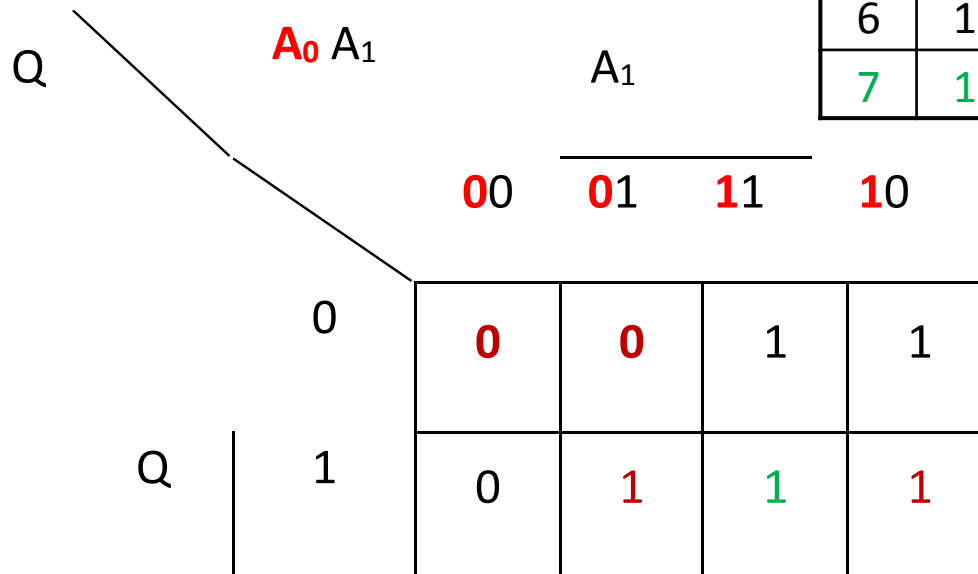
i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0





# Vezérlő tábla kiolvasása

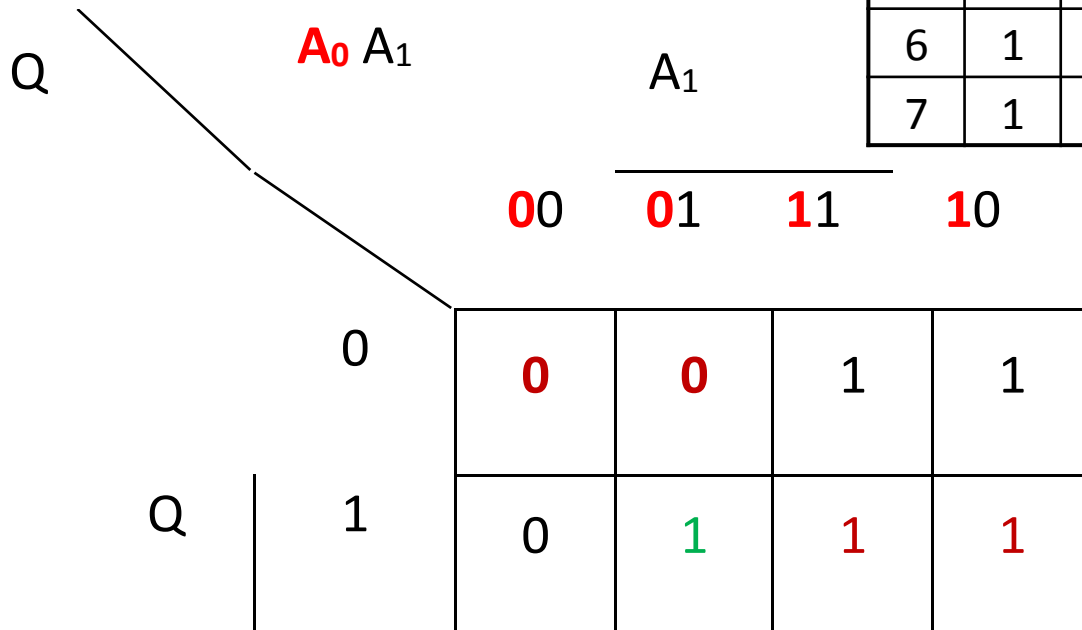
i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0





# Vezérlő tábla kiolvasása

i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0





# Vezérlő tábla kiolvasása

i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0

A<sub>0</sub> A<sub>1</sub>

A<sub>1</sub>

00

01

11

10

Q

0

1

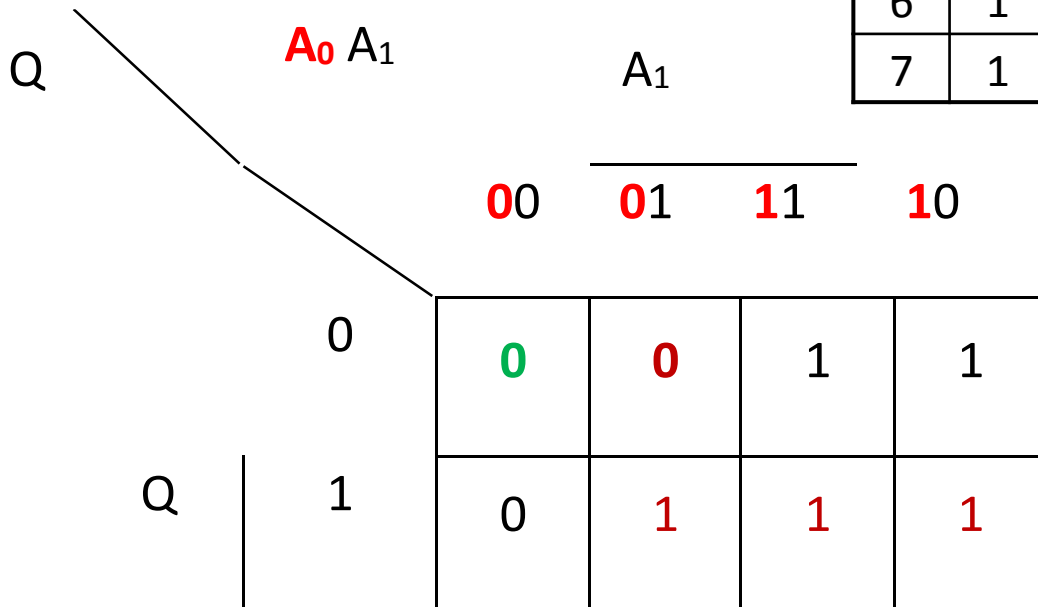
0	0	0	1	1
1	0	1	1	1





# Vezérlő tábla kiolvasása

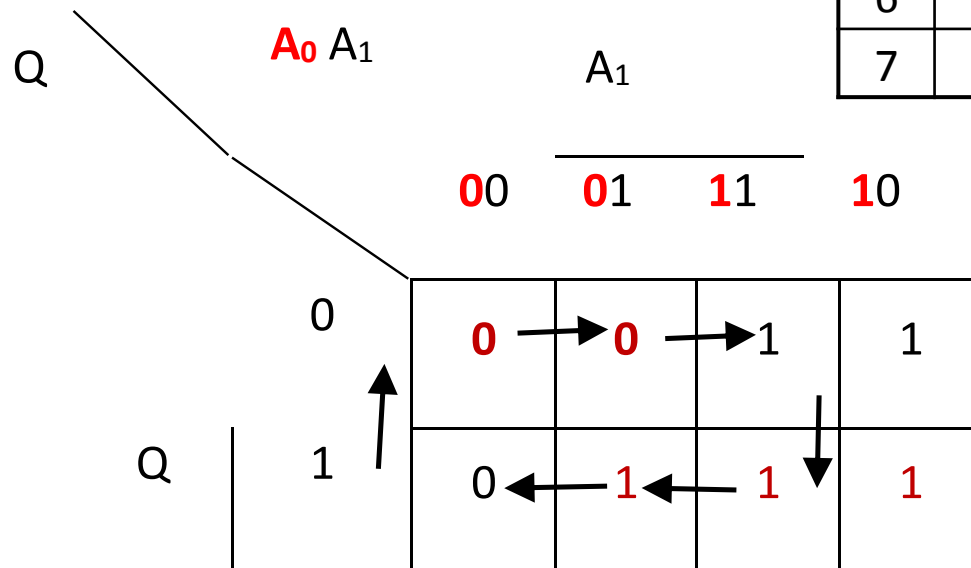
i	Q	A <sub>0</sub>	A <sub>1</sub>	Q <sub>n+1</sub>	Y
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0





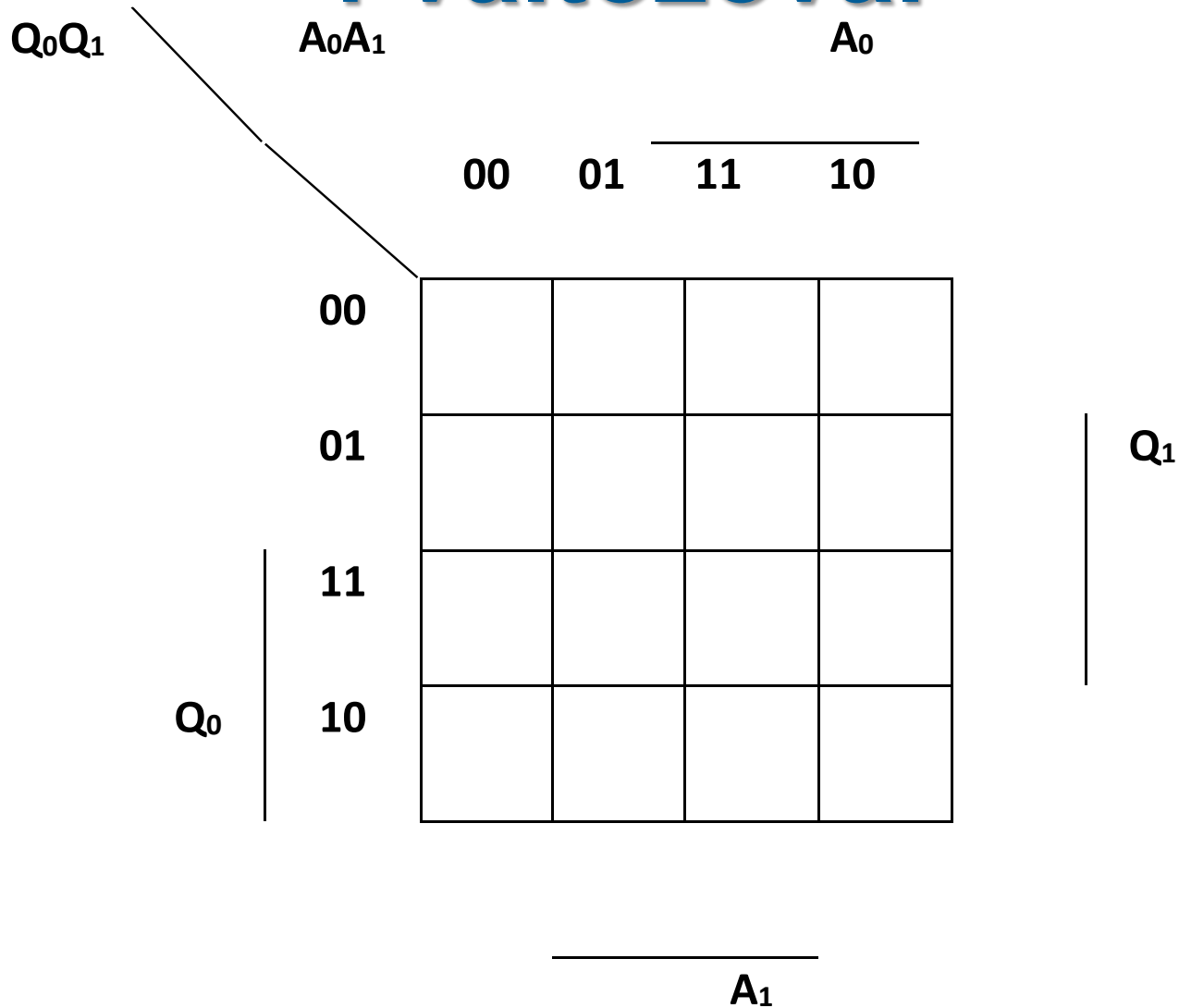
# Vezérlő tábla kiolvasása

$i$	$Q$	$A_0$	$A_1$	$Q_{n+1}$	$Y$
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	1	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0



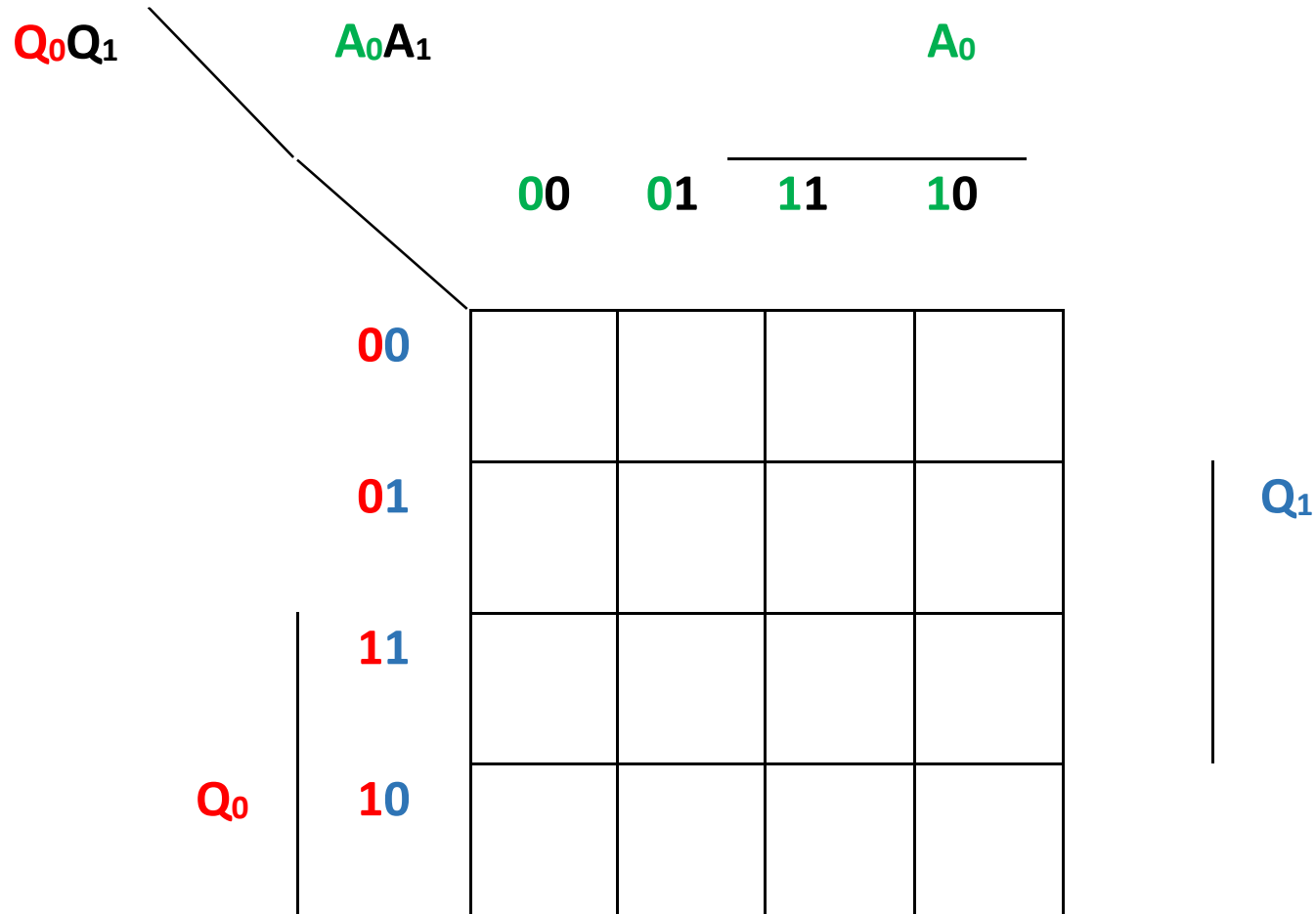


# Vezérlő tábla kiolvasása 4 változóval





# Vezérlő tábla







# Vezérlő tábla

Ó  
B  
U  
D  
A  
I  
  
E  
G  
Y  
E  
T  
E  
M





# Vezérlő tábla

$Q_0Q_1$

$A_0A_1$   $A_0$

00 01 11 10

$Q_0$ 00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
$Q_0$ 10	1000	1001	1011	1010

$Q_1$

$A_1$





# Vezérlő tábla

$Q_0Q_1$

$A_0A_1$

$A_0$

00 01 11 10

00

0

1

3

2

01

4

5

7

6

11

12

13

15

14

$Q_0$

10

8

9

11

10

$Q_1$

Ó  
B  
U  
D  
A  
I  
  
E  
G  
Y  
E  
T  
E  
M





# Vezérlő tábla

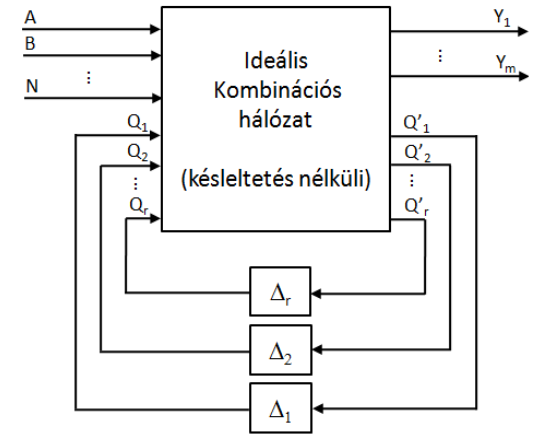
$Q_0Q_1$		$A_0A_1$		$A_0$	
		00	01	11	10
$Q_0$	00	$\overline{Q_0Q_1A_0A_1}$	$\overline{Q_0Q_1A_0A_1}$	$\overline{Q_0Q_1A_0A_1}$	$\overline{Q_0Q_1A_0\overline{A_1}}$
	01	$\overline{Q_0Q_1\overline{A_0}A_1}$	$\overline{Q_0Q_1\overline{A_0}A_1}$	$\overline{Q_0Q_1A_0A_1}$	$\overline{Q_0Q_1A_0\overline{A_1}}$
	11	$Q_0Q_1\overline{A_0}\overline{A_1}$	$Q_0Q_1\overline{A_0}A_1$	$Q_0Q_1A_0A_1$	$Q_0Q_1A_0\overline{A_1}$
	10	$Q_0\overline{Q_1}A_0A_1$	$Q_0\overline{Q_1}A_0A_1$	$Q_0\overline{Q_1}A_0A_1$	$Q_0\overline{Q_1}A_0\overline{A_1}$

$Q_1$





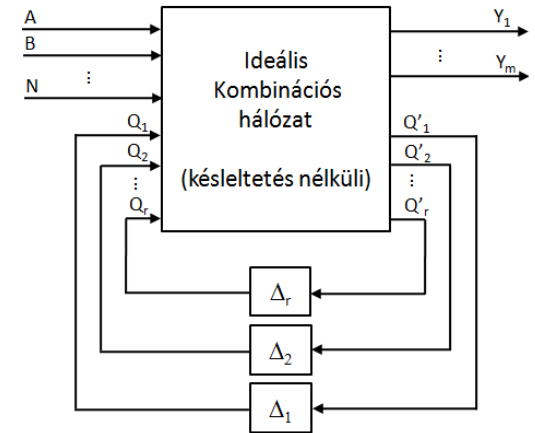
# Példa két belső (szekunder) változóra





# Példa két belső (szekunder) változóra

- ❖ Kiindulás:  $Q_1 Q_0 = A_0 A_1 = 00$  stabil állapot
- ❖ AB bemenet változzon  $A_0 A_1 = 01$ -re
  - ❖ Mindkét belső változó 0-ról 1-re vált
  - ❖ Mi történik, ha eltérő  $\Delta$  késleltetések vannak
- ❖ Általában nem tudjuk előre melyik hurok a gyorsabb
  - ❖ Nem egyértelmű működés



$Q_1 Q_0 \backslash A_0 A_1$	00	01	11	10
00	00	11	01	00
01	11	01	11	00
11	11	11	10	00
10	00	11	00	00

Arrows in the table indicate transitions: a blue arrow from (00,00) to (00,01), a purple arrow from (01,01) to (01,00), and green arrows from (01,01) to (01,11) and from (10,11) to (10,01).

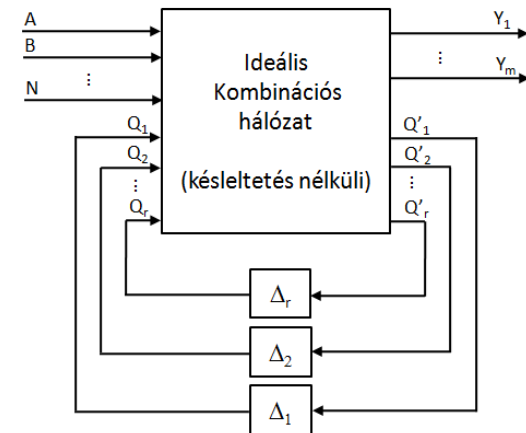
86





# Példa két belső (szekunder) változóra

- ❖ Kiindulás:  $Q_1 Q_0 = A_0 A_1 = 00$  stabil állapot
- ❖ AB bemenet változzon  $A_0 A_1 = 01$ -re
  - ❖ Mindkét belső változó 0-ról 1-re vált
  - ❖ Mi történik, ha eltérő  $\Delta$  késleltetések vannak
- ❖ Általában nem tudjuk előre melyik hurok a gyorsabb
  - ❖ Nem egyértelmű működés



$Q_1 Q_0 \backslash A_0 A_1$	00	01	11	10
00	00	11	01	00
01	11	01	11	00
11	11	11	10	00
10	00	11	00	00

## • Versenyfutás jelenség

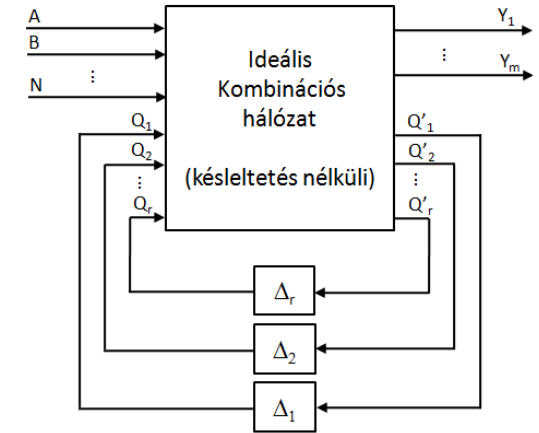
- Ha két szekunder változónak egyszerre kell változnia
- Kritikus versenyfutásról beszélünk, ha az eltérő késleltetések miatt a hálózat eltérő stabil állapotokba kerülhet (példában)
- Azaz ha a vezérlési táblázat versenyfutást tartalmazó oszlopában több stabil állapot is van
- Nem kritikus, ha egy oszlopban csak egy stabil állapot van (utolsó oszlop)





# Példa két belső (szekunder) változóra

- ❖ Kiindulás:  $Q_1 Q_0 = A_0 A_1 = 00$  stabil állapot
- ❖ A bemenet változzon  $A_0 A_1 = 01$ -re
  - ❖ Mindkét belső változó 0-ról 1-re vált
  - ❖ Mi történik, ha eltérő  $\Delta$  késleltetések vannak
- ❖ Általában nem tudjuk előre melyik hurok a gyorsabb
  - ❖ Nem egyértelmű működés



$Q_2 Q_1$ \ $A_0 A_1$	00	01	11	10
00	00	11	01	00
01	11	01	11	00
11	11	11	10	00
10	00	11	00	00

## • Versenyfutás jelenség

- Ha két szekunder változónak egyszerre kell változnia
- Kritikus versenyfutásról beszélünk, ha az eltérő késleltetések miatt a hálózat eltérő stabil állapotokba kerülhet (példában)
- Azaz ha a vezérlési táblázat versenyfutást tartalmazó oszlopában több stabil állapot is van
- Nem kritikus, ha egy oszlopban csak egy stabil állapot van (utolsó oszlop)

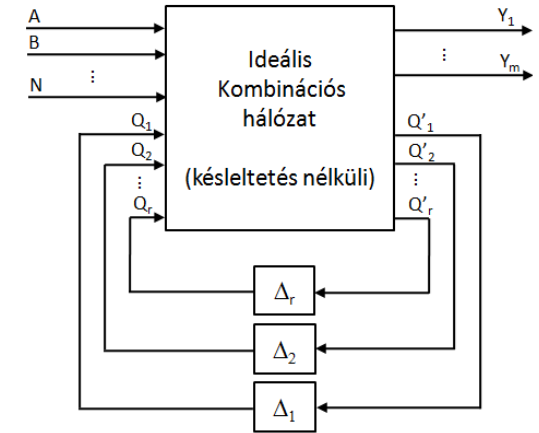






# Példa két belső (szekunder) változóra

- ❖ Kiindulás:  $Q_1 Q_0 = 0$ ;  $A_0 A_1 = 10$  stabil állapot
- ❖ A bemenet változzon  $A_0 A_1 = 11$ -re
  - ❖ Csak az  $Q_0$ -hoz tartozó hurok kell változzon
  - ❖ Nincs versenyfutás de instabil állapot
  - ❖ Újabb és újabb instabil állapotokba lépünk



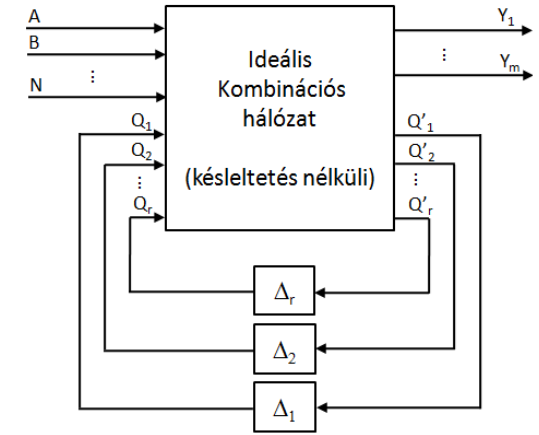
$Q_1 Q_0 \backslash A_0 A_1$	00	01	11	10
00	00	11	01	00
01	11	01	11	00
11	11	11	10	00
10	00	11	00	00





# Példa két belső (szekunder) változóra

- ❖ Kiindulás:  $Q_1 Q_0 = 0$ ;  $A_0 A_1 = 10$  stabil állapot
- ❖ A bemenet változzon  $A_0 A_1 = 11$ -re
  - ❖ Csak az  $Q_0$ -hoz tartozó hurok kell változzon
  - ❖ Nincs versenyfutás de instabil állapot
  - ❖ Újabb és újabb instabil állapotokba lépünk



$Q_1 Q_0 \backslash A_0 A_1$	00	01	11	10
00	00	11	01	00
01	11	01	11	00
11	11	11	10	00
10	00	11	00	00

- Oszcilláció jelenség
  - A hálózat instabil állapotokat vesz fel egymás után
  - Ha nincs stabil állapot a vezérlési tábla egy oszlopában, az adott bemeneti kombináció esetén a rendszer biztosan oszcillálni fog

