

## Az elektronikus hálózatok csoportjai: (jelfeldolgozás alapján)

### ➤ **analóg hálózatok**

- *a jelek értékkészlete (elméletileg) végtelen nagy*
- *a jelek értelmezési tartománya folytonos eloszlású*

### ➤ **digitális hálózatok**

- *a jelek értékkészlete véges*
- *a jelek eloszlása diszkrét*

## A logikai értékek és műveletek

### „Kétértékes” rendszerek:

- Állítások:

IGAZ, HAMIS

- Bináris számrendszer:

1, 0

- Kapcsolók:

bekapcsolva, megszakítva

$\cdot$  („ÉS”)

$+$  („VAGY”)

$$0 \cdot 0 = 0$$

$$0 + 0 = 0$$

$$0 \cdot 1 = 0$$

$$0 + 1 = 1$$

$$1 \cdot 0 = 0$$

$$1 + 0 = 1$$

$$1 \cdot 1 = 1$$

$$1 + 1 = 1$$

(„NEGÁLT”)

$$\overline{0} = 1 \quad \overline{1} = 0$$

- A digitális hálózatokban az információ hordozója a **bit**.*

## A kapcsoló algebra azonosságai (1)

*Asszociativitás:*

$$(A + B) + C = A + (B + C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

*Kommutativitás:*

$$A \cdot B = B \cdot A$$

$$A + B = B + A$$

*Disztributivitás:*

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

*Ellentmondás törvénye:*

$$A \cdot \bar{A} = 0$$

## A kapcsoló algebra azonosságai (2)

*Elnyelési törvények 1.*

$$A + (A \cdot B) = A$$

$$A \cdot (A + B) = A$$

*Elnyelési törvények 2.*

$$A + (\bar{A} \cdot B) = A + B$$

$$A \cdot (\bar{A} + B) = A \cdot B$$

*Elnyelési törvények 3.*

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

*Harmadik kizárásának  
törvénye:*

$$A + \bar{A} = 1$$

## A kapcsoló algebra azonosságai (3)

*Identitások törvénye:*

$$A + 0 = A$$

$$A \cdot 1 = A$$

*Idempotencia törvénye:*

$$A + A = A$$

$$A \cdot A = A$$

*Involúció törvénye:*

$$\bar{\bar{A}} = A$$

*De-Morgan azonosságok:*

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

## KOMBINÁCIÓS HÁLÓZATOK

*A kombinációs hálózat fekete-doboz modellje:*

$X_1 \dots X_n$  :  
bemenetek,  
logikai változók

$Y_1 \dots Y_m$  :  
kimenetek,  
logikai változók



## **A kombinációs hálózat definíciója:**

*a kombinációs hálózat viselkedésének legfontosabb sajátossága, hogy egy meghatározott bemeneti kombináció ismételt rákapcsolásaira - a tranziens idő eltelte után - mindig ugyanazt a kimeneti kombinációt szolgáltatja, függetlenül attól, hogy az adott bemeneti kombináció két rákapcsolása között milyen más bemeneti kombinációkat kapcsoltunk a hálózatra.*

## Kombinációs hálózat definiálása táblázattal

### Példa:

*három bemenet :*

**X1, X2, X3**

*két kimenet:*

**Y1, Y2**

**Bemeneti variációk**

**X1**

**X2**

**X3**

**Kimeneti variációk**

**Y1**

**Y2**

0

0

0

0

1

0

0

1

1

0

0

1

0

0

1

0

1

1

1

1

1

0

0

1

1

1

0

1

0

1

1

1

0

1

1

1

1

1

1

0

## Kombinációs hálózatok specifikációs mélysége

- ***teljesen specifikált („ts”):***

minden bemeneti variációra minden kimenet értéke elő van írva

- ***nem teljesen specifikált („nts”):***

van olyan bemeneti variáció, ahol egy kimeneti változó értéke közömbös

## Logikai függvények megadási módjai:

- *igazságtáblázattal*
- *algebrai kifejezéssel*
- *elvi (grafikus) logikai vázlattal*

- logikai függvény megadása igazságtáblával:***

*Példa: egy teljesen specifikált („ts”), egykimenetű kombinációs hálózat igazságtáblázata*

|                          |                | igazságtábla |          |          |          |
|--------------------------|----------------|--------------|----------|----------|----------|
|                          |                | <b>A</b>     | <b>B</b> | <b>C</b> | <b>F</b> |
| <b><i>bemenetek:</i></b> | <b>A, B, C</b> | <b>0</b>     | <b>0</b> | <b>0</b> | <b>0</b> |
|                          |                | <b>0</b>     | <b>0</b> | <b>1</b> | <b>0</b> |
|                          |                | <b>0</b>     | <b>1</b> | <b>0</b> | <b>1</b> |
|                          |                | <b>0</b>     | <b>1</b> | <b>1</b> | <b>1</b> |
| <b><i>kimenet:</i></b>   | <b>F</b>       | <b>1</b>     | <b>0</b> | <b>0</b> | <b>1</b> |
|                          |                | <b>1</b>     | <b>0</b> | <b>1</b> | <b>0</b> |
|                          |                | <b>1</b>     | <b>1</b> | <b>0</b> | <b>1</b> |
|                          |                | <b>1</b>     | <b>1</b> | <b>1</b> | <b>0</b> |

- *logikai függvény megadása algebrai kifejezéssel (1):*

## **A minterm definíciója:**

*azokat a logikai szorzatokat (termeket), amelyekben a függvény valamennyi változója szerepel, mintermeknek nevezzük. A logikai függvény megadásának ezt a módját, azaz azon mintermek összegét, amelyekhez a függvény „1”-et rendel, mintermes kanonikus normál alaknak nevezzük.*

## • *logikai függvény megadása algebrai kifejezéssel (2):*

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

A hálózat mintermes kanonikus normál alakja:

$$F(A, B, C) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC$$

A szorzat változója

- ponált, ha a bemeneti variációban ,1' szerepel az oszlopában,
- negált, ha a bemeneti variációban ,0' szerepel az oszlopában

$$F(x_1, x_2, \dots, x_n) = \sum_{i=0}^{2^n-1} v_i \cdot m_i$$

$v_i$  : a két bináris érték egyike, 0 vagy 1

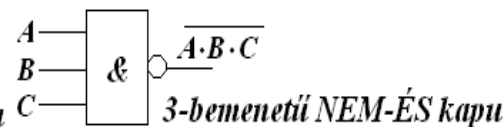
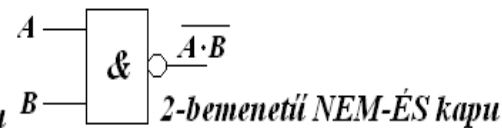
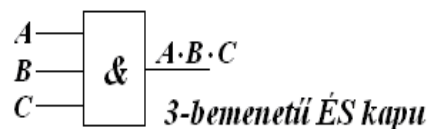
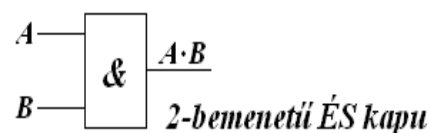
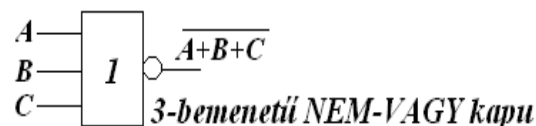
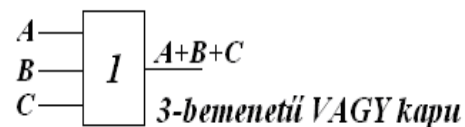
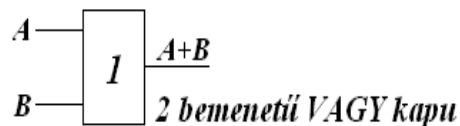
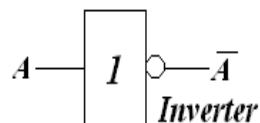
$m_i$  : az igazság-tábla i.sorához rendelt logikai szorzat, amely

- minden változót tartalmaz
- kifejezi a sorhoz tartozó bemeneti kombináció 1-es értékét

$$F(x_1, x_2, \dots, x_n) = \sum_{v_i=1} 1 \cdot m_j + \sum_{v_i=0} 0 \cdot m_k = \sum_{v_i=1} 1 \cdot m_j = \sum_{v_i=1} m_j$$

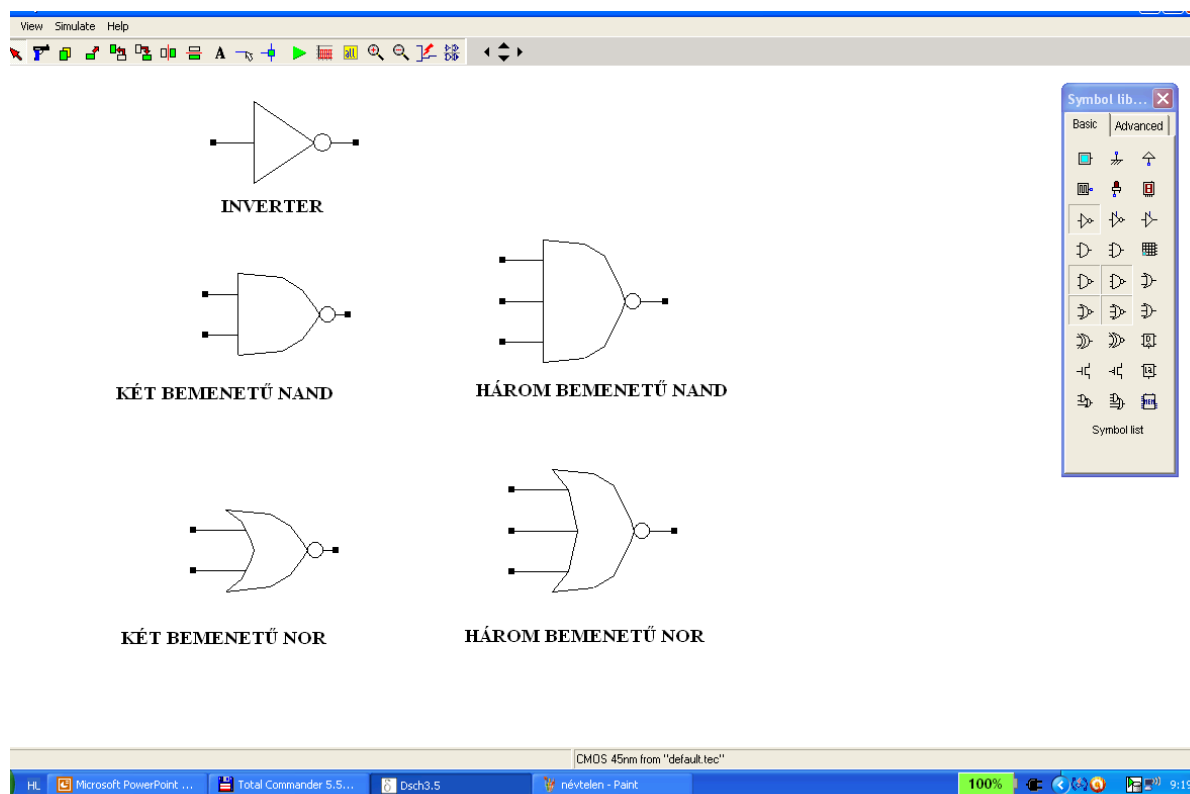
- logikai függvény megadása grafikus logikai vázlattal (1):**

A leggyakrabban használt grafikus logikai szimbólumok (európai szabvány):



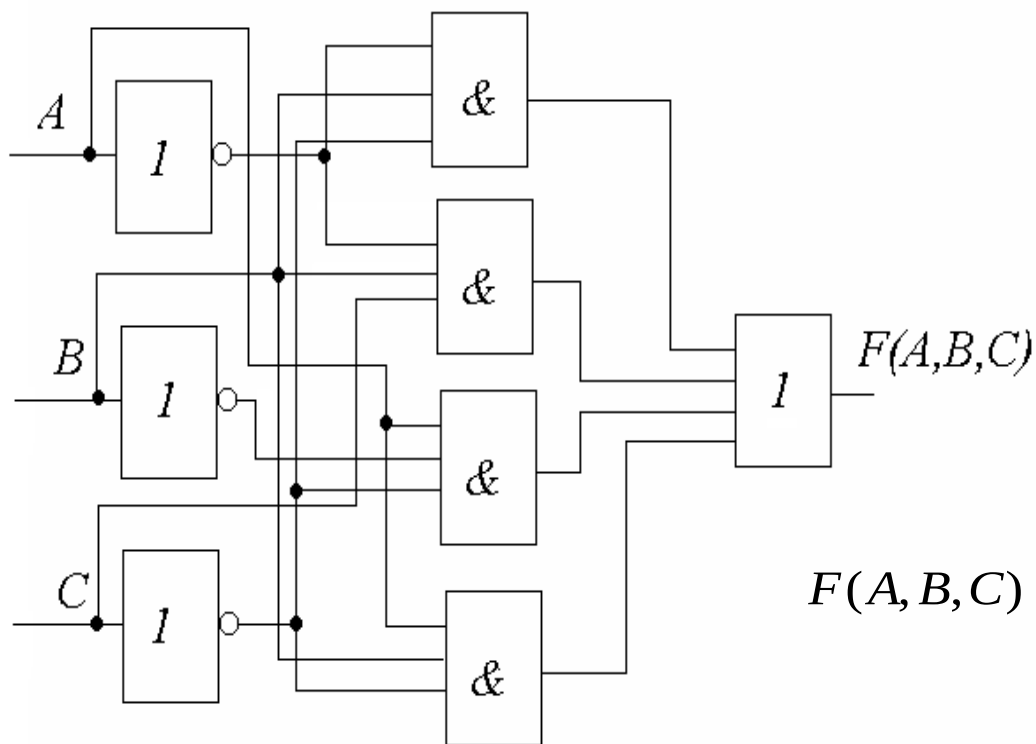
- logikai függvény megadása grafikus logikai vázlattal (2):***

A DSCH szimulátorban használt grafikus logikai szimbólumok (IEEE szabvány):



- logikai függvény megadása grafikus logikai vázlattal (3):***

A mintapélda grafikus logikai szimbólumokkal megadott vázlata:



$$F(A, B, C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC\bar{C}$$

## Nevezetes kétváltozós logikai függvények (1)

| BEM.<br>VÁLT. |    | FÜGGVÉNYÉRTÉKEK |    |    |    |    |    |    |    |    |    |     |     |     |     |     |     |
|---------------|----|-----------------|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|
| x1            | x2 | f0              | f1 | f2 | f3 | f4 | f5 | f6 | f7 | f8 | f9 | f10 | f11 | f12 | f13 | f14 | f15 |
| 0             | 0  | 0               | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1   | 1   | 1   | 1   | 1   | 1   |
| 0             | 1  | 0               | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 0  | 0  | 0   | 0   | 1   | 1   | 1   | 1   |
| 1             | 0  | 0               | 0  | 1  | 1  | 0  | 0  | 1  | 1  | 0  | 0  | 1   | 1   | 0   | 0   | 1   | 1   |
| 1             | 1  | 0               | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0   | 1   | 0   | 1   | 0   | 1   |

## Nevezetes kétváltozós logikai függvények (2)

- **'0'( $F_0$ ) és '1'( $F_{15}$ ) generátor**
- **'ÉS'( $F_1$ ) és 'NEM-ÉS'( $F_{14}$ ) függvény, illetve kapu**
- **'VAGY'( $F_7$ ) és 'NEM-VAGY'( $F_8$ ) függvény, illetve kapu**
- **ANTIVALENCIA( $F_6$ ) és EKVIVALENCIA( $F_9$ ) függvény, illetve kapu**
- **INHIBÍCIÓ( $F_2$ )**
- **IMPLIKÁCIÓ( $F_{13}$ )**

## Logikai függvények egyszerűsítésének módszerei

- *egyszerűsítés algebrai módszerrel*
- *Quine módszere*
- *Karnaugh táblás módszer*

## Logikai függvények egyszerűsítésének módszerei: algebrai módszer (1)

Példa:

*három bemenet :*

A, B, C

*egy kimenet:*

F

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

## Logikai függvények egyszerűsítésének módszerei: algebrai módszer (2)

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

A mintermes kanonikus normál függvényalak egyszerűsítése algebrai módszerrel a disztributivitás, valamint „a harmadik kizárásának törvénye” alapján:

$$F(A, B, C) = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + ABC =$$

$$= \overline{A}B(\overline{C} + C) + A\overline{C}(\overline{B} + B) = \overline{A}B + A\overline{C}$$

## Logikai függvények egyszerűsítésének módszerei: Quine módszer (1)

### Fogalmak

- ***prímimplikáns:*** olyan term, amelyből nem hagyható el több változó (nem egyszerűsíthető tovább)
- ***megkülönböztetett minterm:*** olyan minterm, amelyet csak egy prímimplikáns fed le
- ***lényeges prímimplikáns:*** olyan prímimplikáns, amely legalább egy megkülönböztetett mintermet tartalmaz

## Logikai függvények egyszerűsítésének módszerei: Quine módszer (2)

Példa:

$$F(A, B, C) = \overline{A}\overline{B}\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + ABC$$

1.lépés:

egyszerűsítés

| <i>mintermek</i> | <i>I.</i>                                         | <i>II.</i>                        | <i>III.</i> |
|------------------|---------------------------------------------------|-----------------------------------|-------------|
| 2                | $\overline{A}\overline{B}\overline{C} \checkmark$ | (2, 3) $\overline{A}\overline{B}$ |             |
| 3                | $\overline{A}BC \checkmark$                       | (2, 6) $\overline{B}\overline{C}$ |             |
| 4                | $A\overline{B}\overline{C} \checkmark$            | (4, 6) $A\overline{C}$            |             |
| 6                | $ABC \checkmark$                                  |                                   |             |

## Logikai függvények egyszerűsítésének módszerei: Quine módszer (3)

2.lépés:

feltétlenül szükséges prímisszimplifikánsok kiválasztása

*Prímisszimplifikáns táblázat*

|        |                 | 2 | 3 | 4 | 6 |
|--------|-----------------|---|---|---|---|
| (2, 3) | $\overline{A}B$ | ⊗ | ⊗ |   |   |
| (2, 6) | $B\overline{C}$ | * |   |   | * |
| (4, 6) | $A\overline{C}$ |   |   | ⊗ | ⊗ |

## Logikai függvények egyszerűsítésének módszerei: Quine módszer (4)

### 3.lépés:

redundáns prímimplikáns(ok) kiválasztása és elhagyása, a végleges (egyszerűsített) függvényalak felírása

- redundáns prímimplikáns:  $B\bar{C}$
- irredundáns prímimplikáns:  $\bar{A}B, A\bar{C}$

*A végleges (legegyszerűbb) függvényalak:*

$$F(A, B, C) = \bar{A}B + A\bar{C}$$

## Logikai függvények egyszerűsítésének módszerei: Karnaugh táblás módszer (1)

- kétváltozós Karnaugh tábla:

|          |          |          |          |
|----------|----------|----------|----------|
|          |          | <i>A</i> |          |
|          |          | <i>0</i> | <i>1</i> |
| <i>B</i> | <i>0</i> |          |          |
|          | <i>1</i> |          |          |
|          |          | <i>0</i> | <i>2</i> |
|          |          | <i>1</i> | <i>3</i> |

## Logikai függvények egyszerűsítésének módszerei: Karnaugh táblás módszer (2)

- háromváltozós Karnaugh tábla:

|          |   |          |   |   |   |   |
|----------|---|----------|---|---|---|---|
|          |   | <i>A</i> | 0 | 0 | 1 | 1 |
|          |   | <i>B</i> | 0 | 1 | 1 | 0 |
| <i>C</i> | 0 |          |   |   |   |   |
|          | 1 |          |   |   |   |   |
|          |   |          | 0 | 2 | 6 | 4 |
|          |   |          | 1 | 3 | 7 | 5 |

## Logikai függvények egyszerűsítésének módszerei: Karnaugh táblás módszer (3)

- négyváltozós Karnaugh tábla:

|            |     | <i>A</i> |          |          |          |
|------------|-----|----------|----------|----------|----------|
|            |     | <i>B</i> | <i>B</i> | <i>B</i> | <i>B</i> |
|            |     | 0        | 0        | 1        | 1        |
| <i>C D</i> | 0 0 | 0        | 4        | 12       | 8        |
|            | 0 1 | 1        | 5        | 13       | 9        |
|            | 1 1 | 3        | 7        | 15       | 11       |
|            | 1 0 | 2        | 6        | 14       | 10       |

## Logikai függvények egyszerűsítésének módszerei: Karnaugh táblás módszer (4)

Egyszerűsítés a Karnaugh táblán: szomszédos termek összevonása

**Példa1.:**

|     |     |   |   |    |    |
|-----|-----|---|---|----|----|
|     |     | A |   |    |    |
|     |     | 0 | 0 | 1  | 1  |
|     |     | B |   |    |    |
|     |     | 0 | 1 | 1  | 0  |
| C D | 0 0 |   |   |    |    |
|     |     | 0 | 4 | 12 | 8  |
|     | 0 1 | 1 | 5 | 13 | 9  |
|     |     |   |   |    |    |
| 1 1 |     |   |   |    |    |
|     |     | 3 | 7 | 15 | 11 |
| 1 0 |     |   |   |    |    |
|     |     | 2 | 6 | 14 | 10 |

$B \overline{C} D$

## Logikai függvények egyszerűsítésének módszerei: Karnaugh táblás módszer (5)

Egyszerűsítés a Karnaugh táblán: szomszédos termek összevonása

**Példa2.:**

|     |  |   |  |   |  |
|-----|--|---|--|---|--|
|     |  | A |  |   |  |
|     |  | 0 |  | 1 |  |
| B   |  | 0 |  | 1 |  |
| C D |  | 0 |  | 1 |  |
| 0 0 |  |   |  |   |  |
| 0 1 |  |   |  |   |  |
| 1 1 |  |   |  |   |  |
| 1 0 |  |   |  |   |  |

|   |   |   |    |    |
|---|---|---|----|----|
|   | 0 | 1 | 2  | 3  |
| 0 | 0 | 4 | 12 | 8  |
| 1 | 1 | 5 | 13 | 9  |
| 3 | 1 | 7 | 15 | 11 |
| 2 |   | 6 | 14 | 10 |

*BD* (with arrow pointing to the group of 1s in the 01 and 11 rows, columns 1 and 2)

## Logikai függvények egyszerűsítésének módszerei: Karnaugh táblás módszer (6)

Teljesen specifikált hálózatok egyszerűsítése a Karnaugh táblán: szomszédos termek összevonása, prímmimplikáns képzés

**Példa:**

|   |   |   |   |   |
|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 1 | 0 |
| C |   |   |   |   |
| 0 |   | 1 | 1 | 1 |
| 1 |   | 1 |   |   |

prímmimplikánsok:

$$\overline{A}B, B\overline{C}, A\overline{C}$$

↑  
redundáns prímmimplikáns

$$F(A, B, C) = \overline{A}B + A\overline{C}$$

## Logikai függvények egyszerűsítésének módszerei: Karnaugh táblás módszer (7)

Nem teljesen specifikált hálózatok egyszerűsítése a Karnaugh táblán: szomszédos termek összevonása, prímmimplikáns képzés

**Példa:**

|                    |    |    |    |    |
|--------------------|----|----|----|----|
| $CD \backslash AB$ | 00 | 01 | 11 | 10 |
|                    | 00 | 01 | 11 | 10 |
| 00                 |    |    |    | 1  |
| 01                 |    | 1  | 1  | —  |
| 11                 | —  | —  | —  | —  |
| 10                 |    |    |    |    |

prímmimplikánsok:

$$BD, \quad A\bar{C}D, \quad A\bar{B}\bar{C}$$



redundáns prímmimplikáns

$$F(A, B, C) = BD + A\bar{B}\bar{C}$$

## Egy- és többkimenetű kombinációs hálózatok tervezése

- Egykimenetű kombinációs hálózatok tervezési lépései:
1. *igazságtábla és/vagy Karnaugh tábla megadása,*
  2. *függvény kiolvasás és egyszerűsítés a Karnaugh táblán (a minimalizálás céljából),*
  3. *a rendelkezésre álló logikai építőelemekhez igazodó függvényalak felírás (átalakítás),*
  4. *kapusintű realizáció a rendelkezésre álló logikai építőelemek segítségével.*

## Teljesen specifikált, egykimenetű kombinációs hálózatok tervezése - mintapélda

Tervezzük meg „NEM-ÉS” (NAND) kapukkal a következő specifikációval megadott négybemenetű (A,B,C,D) és egykimenetű (F), teljesen specifikált kombinációs hálózatot:

$$F('1') : ( 2, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15) !$$

A bemenetek sorrendje: ABCD (MSB!)

## Teljesen specifikált, egykimenetű kombinációs hálózatok tervezése - mintapélda

Megoldás:

1.lépés: a Karnaugh tábla felvétele, és a megadott specifikáció szerinti kitöltése

|                   |   | <i>A</i> 0 |   | 0 | 1 | 1 |
|-------------------|---|------------|---|---|---|---|
|                   |   | <i>B</i> 0 |   | 1 | 1 | 0 |
| <i>C</i> <i>D</i> |   |            |   |   |   |   |
| 0 0               |   |            | 1 | 1 |   |   |
| 0 1               |   |            | 1 | 1 | 1 |   |
| 1 1               |   |            |   | 1 | 1 |   |
| 1 0               | 1 | 1          | 1 | 1 |   |   |

## Teljesen specifikált, egykimenetű kombinációs hálózatok tervezése - mintapélda

Megoldás:

2.lépés: függvény kiolvasás és egyszerűsítés a Karnaugh táblán (a minimalizálás céljából),

|     |   |     |   |   |   |   |
|-----|---|-----|---|---|---|---|
|     |   | A 0 |   | 0 | 1 | 1 |
|     |   | B 0 |   | 1 | 1 | 0 |
| C D |   |     |   |   |   |   |
| 0 0 |   |     | 1 | 1 |   |   |
| 0 1 |   |     | 1 | 1 | 1 |   |
| 1 1 |   |     |   | 1 | 1 |   |
| 1 0 | 1 | 1   | 1 | 1 |   |   |

*prímimplikánsok:*

$B\bar{D}$ ,  $\bar{B}\bar{C}$ ,  $AD$ ,  $AC$ ,  $C\bar{D}$ ,  $AB$

*irredundáns, szükséges  
és elégséges lefedés:*

$\bar{B}\bar{C}$ ,  $AD$ ,  $C\bar{D}$

## Teljesen specifikált, egykimenetű kombinációs hálózatok tervezése - mintapélda

Megoldás:

3.lépés: a rendelkezésre álló logikai építőelemekhez igazodó (mintermes)függvényalak felírás (átalakítás)

*Az irredundáns lefedés  
prímimplikánsai:*

$$\overline{BC}, \quad AD, \quad \overline{CD}$$

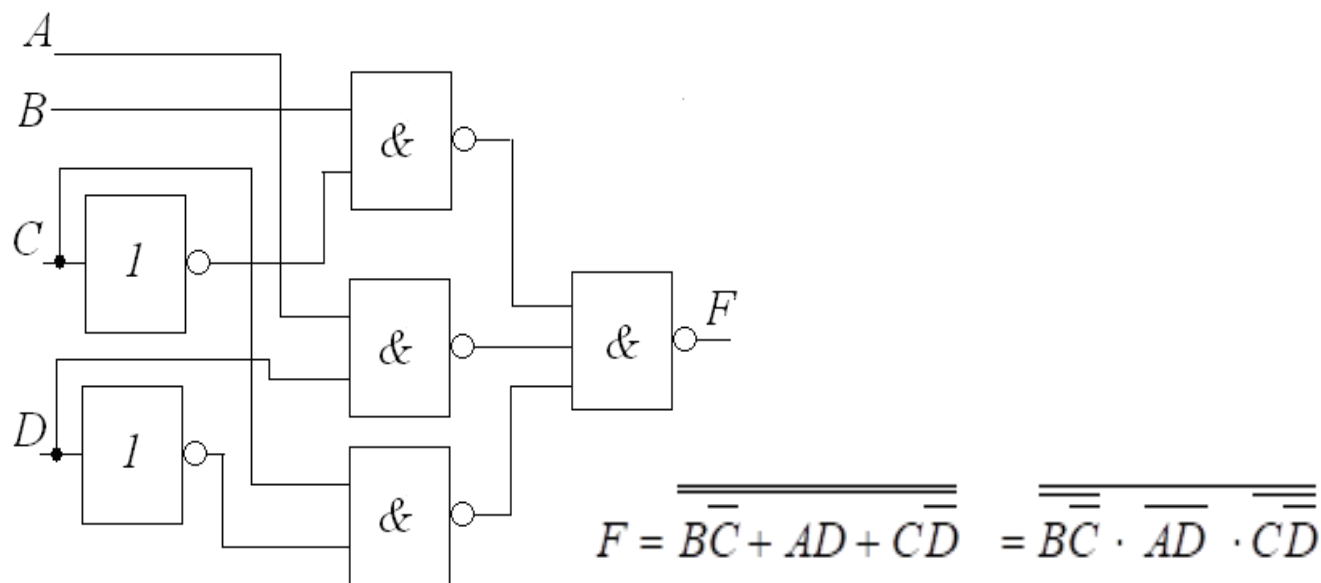


$$F = \overline{\overline{BC} + AD + \overline{CD}} = \overline{\overline{BC}} \cdot \overline{AD} \cdot \overline{\overline{CD}}$$

## Teljesen specifikált, egykimenetű kombinációs hálózatok tervezése - mintapélda

Megoldás:

4.lépés: kapuszentű realizáció a rendelkezésre álló logikai építőelemek segítségével



## Nem teljesen specifikált, egykimenetű kombinációs hálózatok tervezése – mintapélda1.

Tervezzük meg „NEM-ÉS” (NAND) kapukkal a következő specifikációval megadott négybemenetű (A,B,C,D) és egykimenetű (F), nem teljesen specifikált kombinációs hálózatot:

$$F('1') : ( 2, 4, 5, 9, 10, 11, 12, 14, 15)$$

$$F('') : (0, 6, 13)$$

A bemenetek sorrendje: ABCD (MSB!)

## Nem teljesen specifikált, egykimenetű kombinációs hálózatok tervezése – mintapélda1.

Megoldás:

1.lépés: a Karnaugh tábla felvétele, és a megadott specifikáció szerinti kitöltése

|   |   | A |   |   |   |
|---|---|---|---|---|---|
|   |   | 0 | 0 | 1 | 1 |
| C | B | 0 | 1 | 1 | 0 |
|   | D |   |   |   |   |
| 0 | 0 | – | 1 | 1 |   |
| 0 | 1 |   | 1 | – | 1 |
| 1 | 1 |   |   | 1 | 1 |
| 1 | 0 | 1 | – | 1 | 1 |

## Nem teljesen specifikált, egykimenetű kombinációs hálózatok tervezése – mintapélda1.

Megoldás:

2.lépés: függvény kiolvasás és egyszerűsítés a Karnaugh táblán (a minimalizálás céljából),

|   |   | A |   |   |   |
|---|---|---|---|---|---|
|   |   | 0 | 0 | 1 | 1 |
| C | B | 0 | 1 | 1 | 0 |
|   | D |   |   |   |   |
| 0 | 0 | - | 1 | 1 |   |
| 0 | 1 |   | 1 | - | 1 |
| 1 | 1 |   |   | 1 | 1 |
| 1 | 0 | 1 | - | 1 | 1 |

*prímimplikánsok:*

$\overline{A}\overline{D}$ ,  $B\overline{D}$ ,  $B\overline{C}$ ,  $AD$ ,  $AC$ ,  $C\overline{D}$ ,  $AB$

*irredundáns, szükséges és elégséges lefedés:*

$B\overline{C}$ ,  $AD$ ,  $C\overline{D}$

## Nem teljesen specifikált, egykimenetű kombinációs hálózatok tervezése – mintapélda1.

Megoldás:

3.lépés: a rendelkezésre álló logikai építőelemekhez igazodó (mintermes)függvényalak felírás (átalakítás)

*Az irredundáns lefedés  
prímimplikánsai:*

$$\overline{BC}, \quad AD, \quad \overline{CD}$$

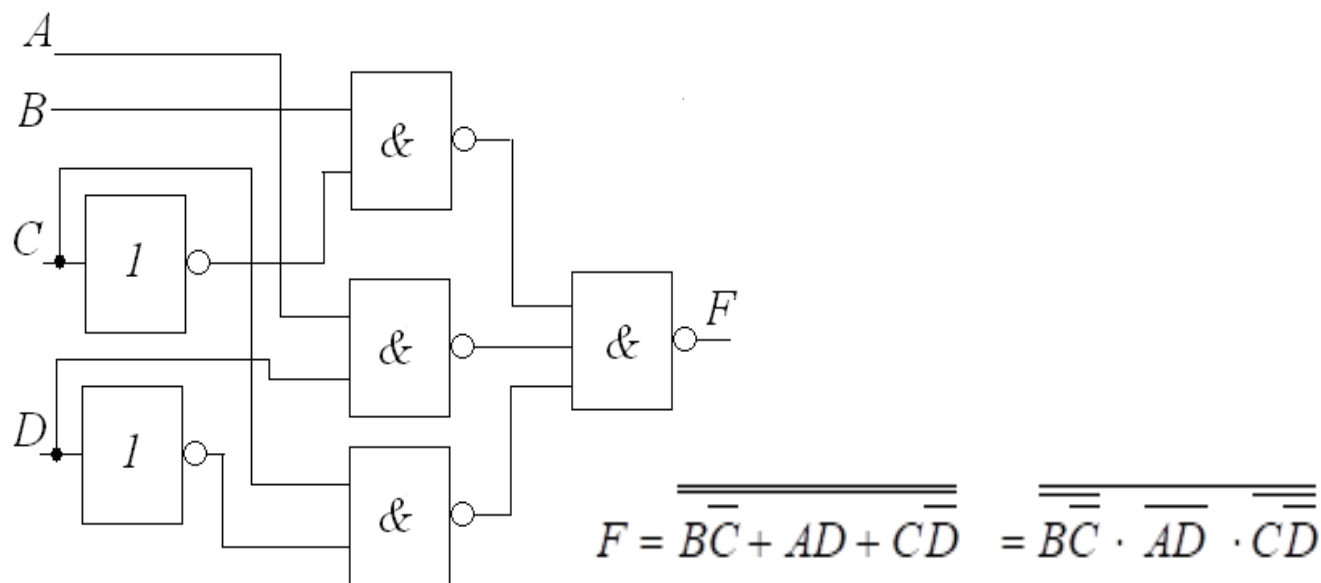


$$F = \overline{\overline{\overline{BC}} + \overline{\overline{AD}} + \overline{\overline{CD}}} = \overline{\overline{\overline{BC}} \cdot \overline{\overline{AD}} \cdot \overline{\overline{CD}}}$$

## Nem teljesen specifikált, egykimenetű kombinációs hálózatok tervezése – mintapélda1.

Megoldás:

4.lépés: kapuszentű realizáció a rendelkezésre álló logikai építőelemek segítségével



## Nem teljesen specifikált, egykimenetű kombinációs hálózatok tervezése – mintapélda2.

Példa egy, az alábbi táblázat által specifikált hálózat függvény minimalizálása céljából elvégzett minterm összevonás bemutatására:

|   |   | A |   |   |   |
|---|---|---|---|---|---|
|   |   | 0 | 0 | 1 | 1 |
| C | B | 0 | 1 | 1 | 0 |
|   | D |   |   |   |   |
| 0 | 0 | 1 | 1 |   | – |
| 0 | 1 | 1 |   |   | – |
| 1 | 1 |   |   |   |   |
| 1 | 0 | 1 |   |   | 1 |

Az irredundáns lefedéshez szükséges prímmimplikánsok:

$$\overline{C}\overline{B}, \quad \overline{A}\overline{C}\overline{D}, \quad \overline{B}\overline{D}$$

## ➤ Többkimenetű kombinációs hálózatok tervezési lépései:

1. *Karnaugh tábla segítségével felírandó valamennyi kimenethez rendelt függvény príimplikánsa,*
2. *az összes lehetséges függvény-szorzat príimplikánsainak megadása,*
3. *az egyes kimeneti függvények mintermjeinek lefedése a saját, más kimenetekhez nem tartozó príimplikánsokkal, illetve azokkal a maximális közös implikánsokkal, amelyek az adott függvénynek implikánsai*

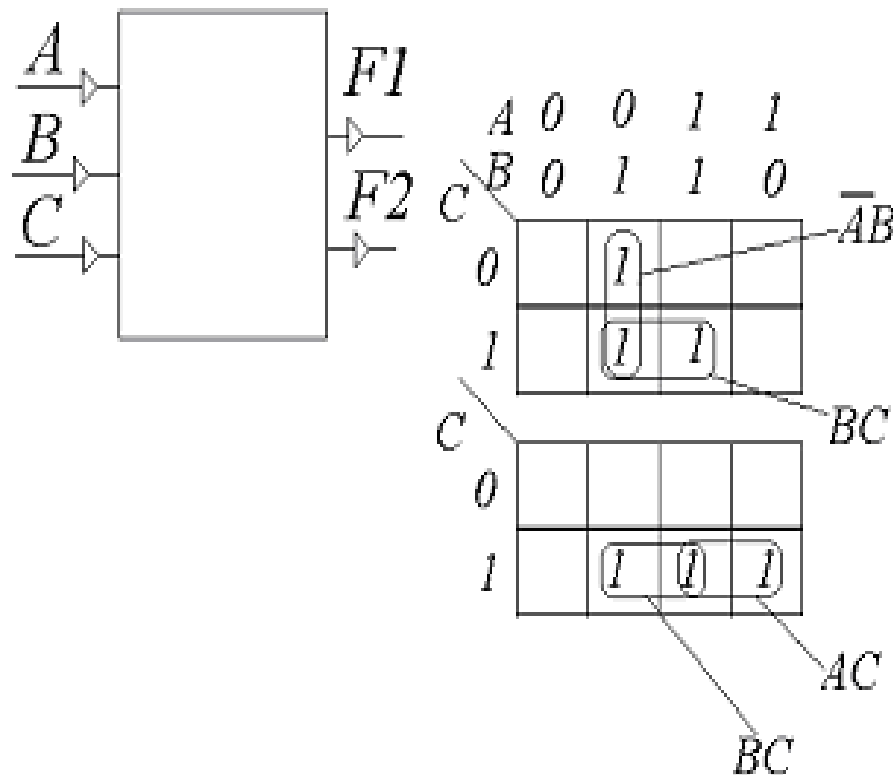
## Többkimenetű kombinációs hálózatok tervezése – mintapélda1.

Adott egy három( $A, B, C$ ) bemenetű és két( $F1, F2$ ) kimenetű kombinációs hálózat az alábbi Karnaugh táblák szerinti specifikációban.

Tervezzük meg a hálózatot a legegyszerűbb változatban!

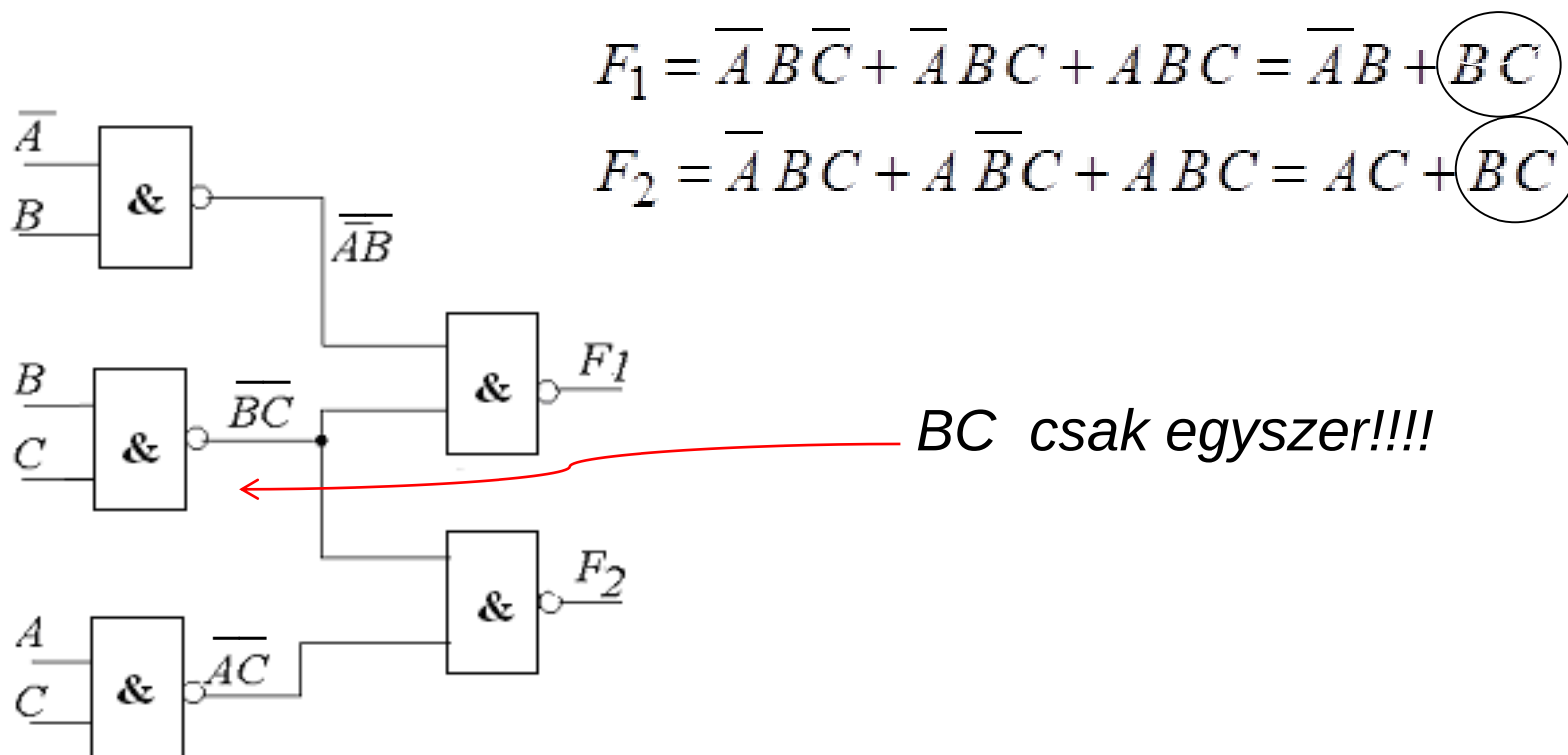
A bemenetek sorrendje: ABCD (MSB!)

## Többkimenetű kombinációs hálózatok tervezése – mintapélda1.



## Többkimenetű kombinációs hálózatok tervezése – mintapélda1.

Megoldás (közös prímplikáns keresés):



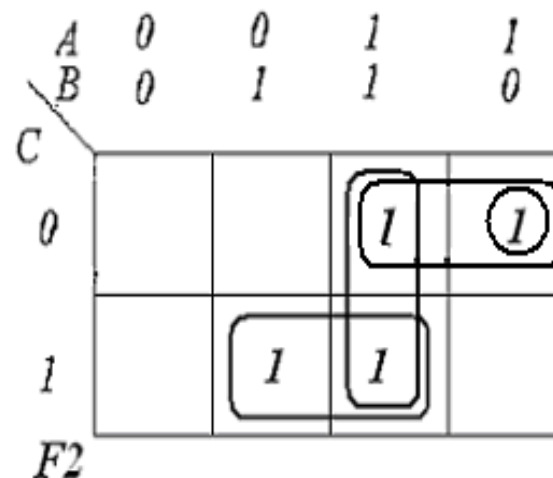
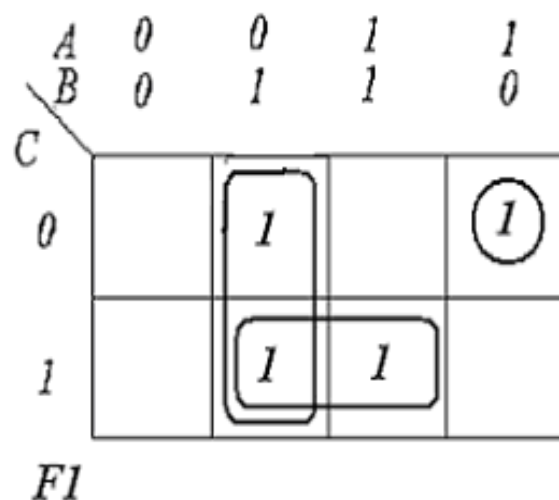
## Többkimenetű kombinációs hálózatok tervezése – mintapélda2.

Adott egy három( $A, B, C$ ) bemenetű és két( $F1, F2$ ) kimenetű kombinációs hálózat az alábbi Karnaugh táblák szerinti specifikációban!

Tervezzük meg a hálózatot a legegyszerűbb változatban!

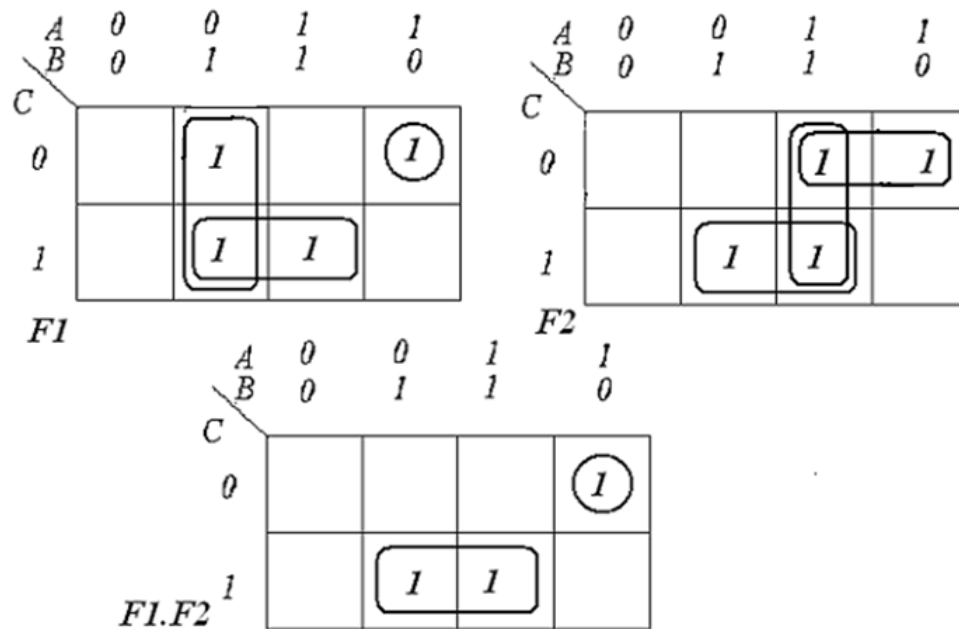
A bemenetek sorrendje: ABCD (MSB!)

## Többkimenetű kombinációs hálózatok tervezése – mintapélda2.



## Többkimenetű kombinációs hálózatok tervezése – mintapélda2.

Megoldás (közös implikáns keresés):



Függvény-szorzat  
prímimplikánsainak előállítása:

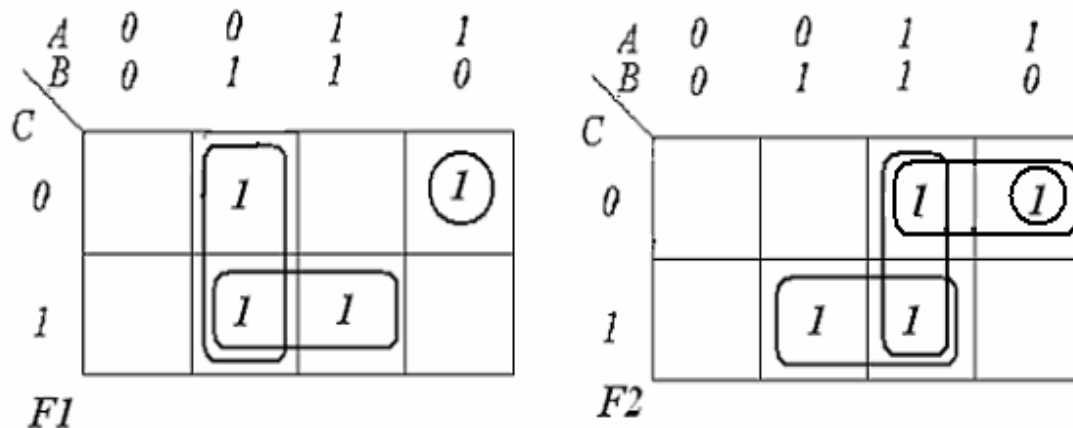
$$F_1 : \bar{A}B, BC, A\bar{B}\bar{C}$$

$$F_2 : BC, AB, A\bar{C}$$

$$F_1 \cdot F_2 : BC, A\bar{B}\bar{C}$$

## Többkimenetű kombinációs hálózatok tervezése – mintapélda2.

Megoldás (közös implikáns keresés):



$A \bar{C}$  helyett  $A \bar{B} \bar{C} !!$

## Hazárdok kombinációs hálózatokban

*A hazárdok azok az eltérések az ideális, késleltetés nélküli hálózatok viselkedésétől, amelyek a logikai kapuk időbeli késleltetéséből adódnak.*

### ❖ *Típusai*

- *Statikus hazárdok*
- *Dinamikus hazárdok*
- *Funkcionális hazárdok*
- *Lényeges hazárdok (későbbi fejezet)*

## ■ Statikus hazard

*Ha egyetlen bemeneti változó logikai értékének megváltozásakor a kimenet a specifikáció szerint nem változna, de a realizált hálózat kimenetén mégis átmeneti változás zajlik le, statikus hazardról beszélünk.*

### ❖ *Típusai*

- *,0'-ás típusú statikus hazard*
- *,1'-es típusú statikus hazard*

## ,0'-ás típusú statikus hazard:

- a specifikált hálózat kimenete a bemeneti változás ellenére alacsony logikai szinten ('0') kell, hogy maradjon, de a realizált hálózat - átmenetileg - egy magas, '1'-es logikai szintű impulzust mutat.

## ,1'-es típusú statikus hazard:

- a specifikált hálózat kimenete a bemeneti változás ellenére magas logikai szinten ('1') kell, hogy maradjon, de a realizált hálózat - átmenetileg - egy alacsony, '0'-ás logikai szintű impulzust mutat.

## Statikus hazárd keletkezése (1)

### Példa

Vizsgáljuk meg az alábbi, grafikus sémával megadott kombinációs hálózat kapuszentű viselkedését az  $ABC$  bemenetek  $1\ 1\ 0 \rightarrow 0\ 1\ 0$  bitkombináció változása esetén!

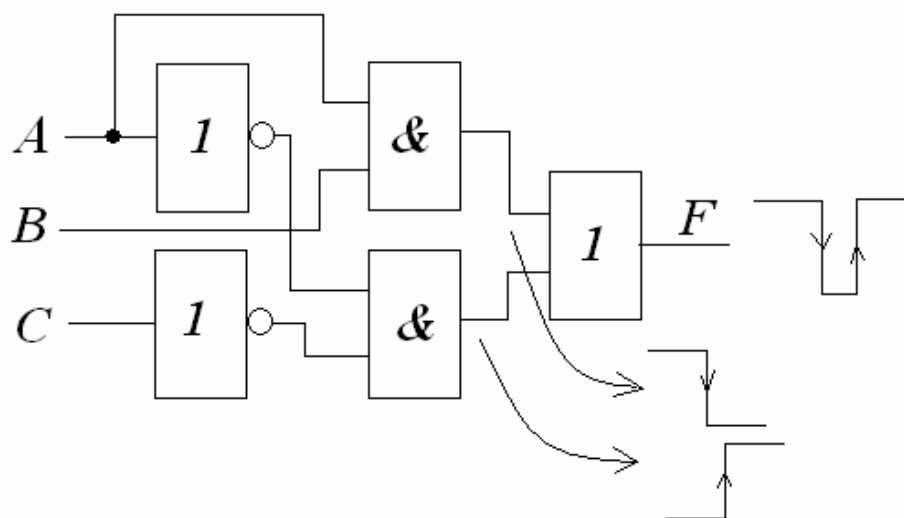
|     |     |                 |   |   |  |
|-----|-----|-----------------|---|---|--|
|     |     | $A\ 0\ 0\ 1\ 1$ |   |   |  |
|     |     | $B\ 0\ 1\ 1\ 0$ |   |   |  |
| $C$ | $0$ | 1               | 1 | 1 |  |
|     | $1$ |                 |   | 1 |  |

$$F(A,B,C) = \overline{A}\overline{C} + AB$$

## Statikus hazárd keletkezése (2)

|          |          |   |   |   |   |
|----------|----------|---|---|---|---|
|          | <i>A</i> | 0 | 0 | 1 | 1 |
|          | <i>B</i> | 0 | 1 | 1 | 0 |
| <i>C</i> | 0        | 1 | 1 | 1 |   |
|          | 1        |   |   | 1 |   |

$$F(A,B,C) = \overline{A}\overline{C} + AB$$



,1'-es típusú statikus hazárd

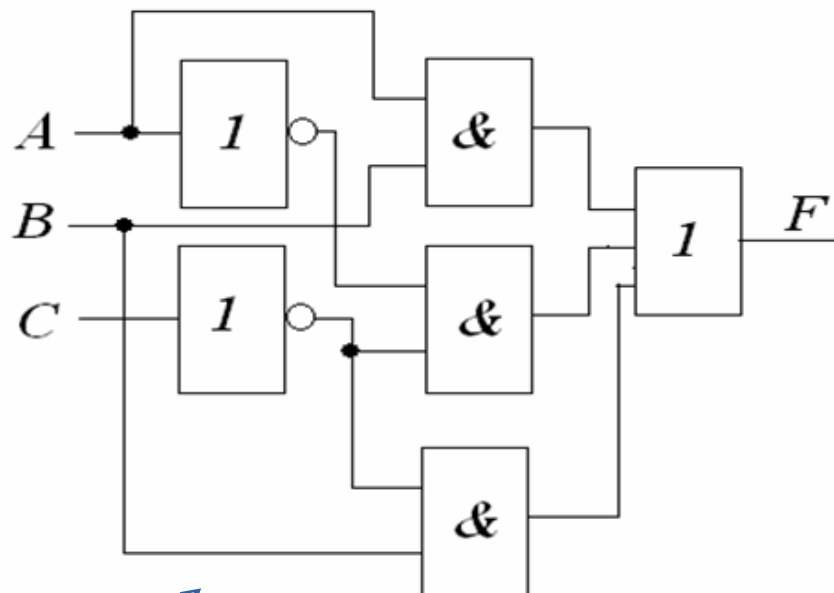
## Statikus hazard kiküszöbölése

Megoldás: redundáns (,átfedő') prímisszorzók alkalmazása a kritikus átmenet helyén

|   |   |   |   |   |   |
|---|---|---|---|---|---|
|   | A | 0 | 0 | 1 | 1 |
|   | B | 0 | 1 | 1 | 0 |
| C | 0 | 1 | 1 | 1 |   |
| 1 |   |   |   | 1 |   |

$$F(A, B, C) = \bar{A}\bar{C} + AB + \bar{B}\bar{C}$$

redundáns term, de megszünteti a hazardot



- **Dinamikus hazard:**

az adott hálózat kimenetének szintet *kell* váltania, de ezt kétszer teszi.

*Kiküszöbölés:* a statikus hazardok megszüntetésével.

- **Funkcionális hazard:**

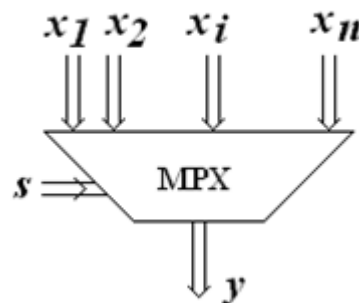
több bemeneti változó együttes változásakor a kimeneten vagy a specifikációtól eltérő szintváltás, vagy többszörös szintváltás jelentkezik.

*Kiküszöbölés:* szinkronizációval.

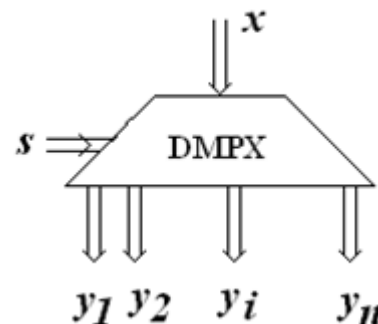
## Programozható kombinációs hálózatok (1)

*Az adatút kijelölő eszközök, mint programozható kombinációs hálózatok:*

- *multiplexerek (MPX,MUX)*



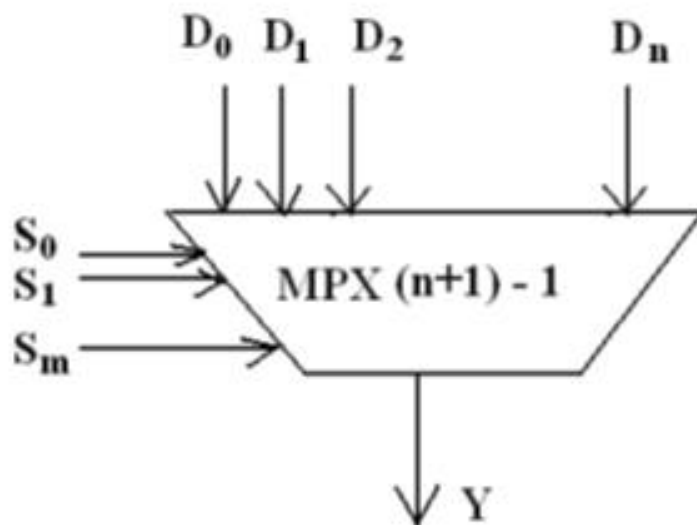
- *demultiplexerek (DMPX,DEMUX)*



## Programozható kombinációs hálózatok (2):

logikai függvények megvalósítása bit-szervezésű  
multiplexerekkel

*Általános séma:*



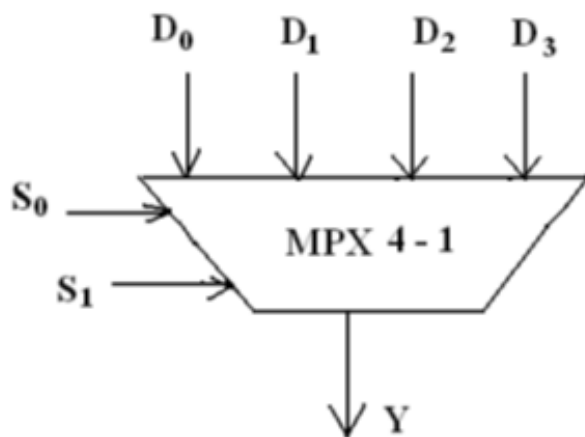
$n + 1 : 2$  egész számú hatványa

$$m = \lceil \log_2 (n + 1) \rceil$$

## Programozható kombinációs hálózatok (3):

logikai függvények megvalósítása bit-szervezésű  
multiplexerekkel

*Példa: MPX4-1 multiplexer sémája, és a működést leíró igazságtáblája*

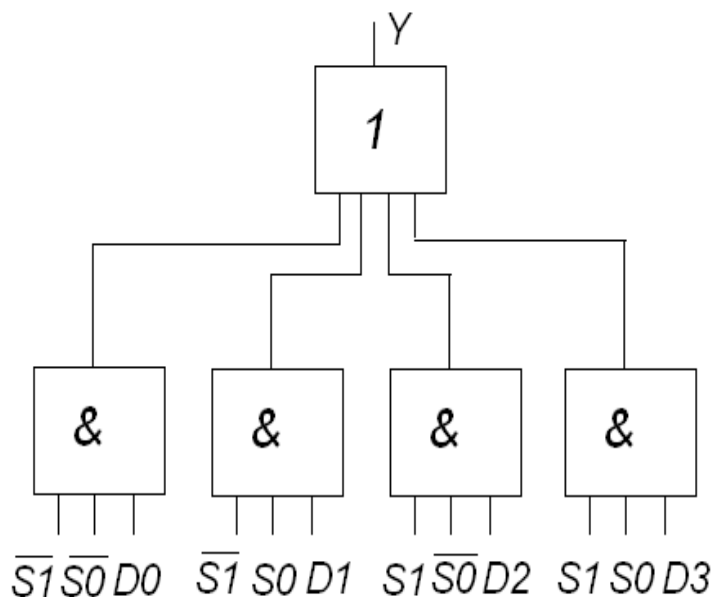


| S1 | S0 | Y              |
|----|----|----------------|
| 0  | 0  | D <sub>0</sub> |
| 0  | 1  | D <sub>1</sub> |
| 1  | 0  | D <sub>2</sub> |
| 1  | 1  | D <sub>3</sub> |

## Programozható kombinációs hálózatok (4):

logikai függvények megvalósítása bit-szervezésű  
multiplexerekkel

*Példa: az MPX4-1 multiplexer kapusintű felépítése*



## **Programozható kombinációs hálózatok (5):**

logikai függvények megvalósítása bit-szervezésű  
multiplexerekkel

### **A multiplexer, mint programozható egykimenetű kombinációs hálózat:**

a függvény mintermjeit a címző bemenetekre adott címek képviselik, és a megcímzett adatbemenetre rá kell kapcsolnunk az adott mintermhez tartozó logikai értéket.

Ezeknek a logikai konstansoknak a bemenetekre való kapcsolását a multiplexer programozásának tekinthetjük.

## Programozható kombinációs hálózatok (6):

logikai függvények megvalósítása bit-szervezésű  
multiplexerekkel

Példa:

Valósítsuk meg multiplexer segítségével (programozásával) az EXOR függvény kapus szintű realizációját!

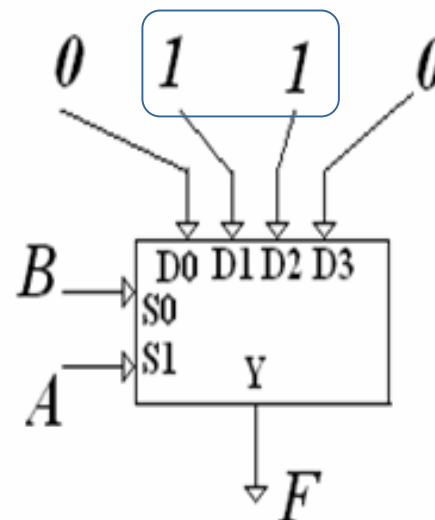
$$F_{(EXOR)} = \bar{A}B + A\bar{B}$$

## Programozható kombinációs hálózatok (7):

logikai függvények megvalósítása bit-szervezésű  
multiplexerekkel

Megoldás:

| <i>A</i> | <i>B</i> | <i>F</i> |      |
|----------|----------|----------|------|
| 0        | 0        | 0        | (D0) |
| 0        | 1        | 1        | (D1) |
| 1        | 0        | 1        | (D2) |
| 1        | 1        | 0        | (D3) |

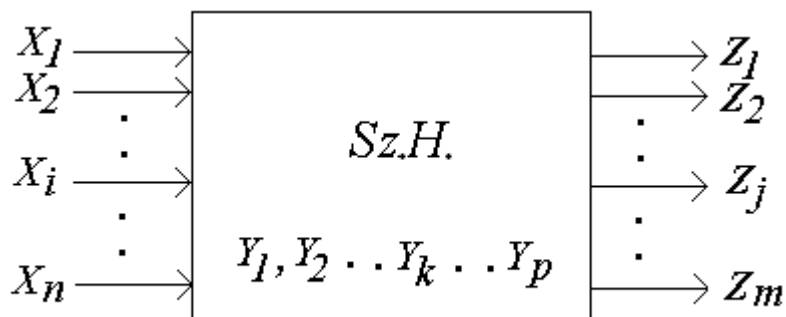


$$F_{(EXOR)} = \bar{A}B + A\bar{B}$$

## Sorrendi (szekvenciális) hálózatok

*Definíció: a sorrendi (szekvenciális) hálózatok legfontosabb tulajdonsága, hogy a kimeneti érték (kombináció) nem csak a pillanatnyi bemeneti értéktől (kombinációtól) függ, hanem a korábbi bemeneti értéktől (kombinációtól), sőt azok sorrendjétől is.*

## A sorrendi hálózatok fekete-doboz modellje (1)



$$f_z : \mathbf{X} \times \mathbf{Y} \Rightarrow \mathbf{Z}$$

$$f_y : \mathbf{X} \times \mathbf{Y} \Rightarrow \mathbf{Y}$$

$\mathbf{X}$ : a bemenet kombinációk halmaza

$\mathbf{Z}$ : a kimeneti kombinációk halmaza

$\mathbf{Y}$ : a szekundér változók halmaza

$$Z_1 = f_{z1}(X_1, \dots, X_i, \dots, X_n, Y_1, \dots, Y_k, \dots, Y_p)$$

$$Z_2 = f_{z2}(X_1, \dots, X_i, \dots, X_n, Y_1, \dots, Y_k, \dots, Y_p)$$

$$\dots \dots \dots$$

$$Z_j = f_{zj}(X_1, \dots, X_i, \dots, X_n, Y_1, \dots, Y_k, \dots, Y_p)$$

$$\dots \dots \dots$$

$$Z_m = f_{zm}(X_1, \dots, X_i, \dots, X_n, Y_1, \dots, Y_k, \dots, Y_p)$$

$$Y'_1 = f_{y1}(X_1, \dots, X_i, \dots, X_n, Y_1, \dots, Y_k, \dots, Y_p)$$

$$Y'_2 = f_{y2}(X_1, \dots, X_i, \dots, X_n, Y_1, \dots, Y_k, \dots, Y_p)$$

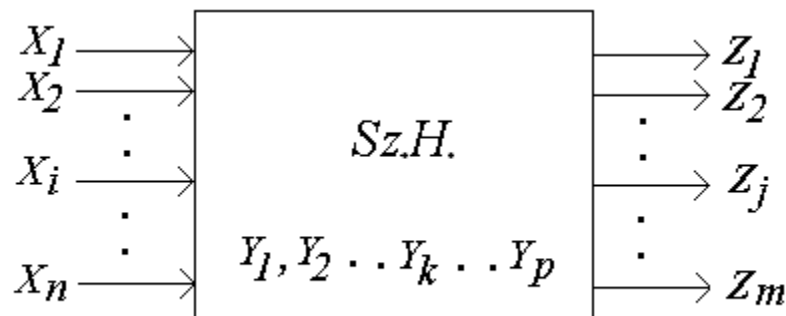
$$\dots \dots \dots$$

$$Y'_k = f_{yk}(X_1, \dots, X_i, \dots, X_n, Y_1, \dots, Y_k, \dots, Y_p)$$

$$\dots \dots \dots$$

$$Y'_p = f_{yp}(X_1, \dots, X_i, \dots, X_n, Y_1, \dots, Y_k, \dots, Y_p)$$

## A sorrendi hálózatok fekete-doboz modellje (2)



$$f_y : (x_i^t, y_k^t) \rightarrow y_l^{t+\Delta t}$$

$$f_z : (x_i^t, y_k^t) \rightarrow z_j^t$$

$x_i^t$  : egy bemeneti kombináció  $t$  pillanatban

$z_j^t$  : egy kimeneti kombináció  $t$  pillanatban

$y_k^t$  : egy szekundér változó kombináció  $t$  pillanatban

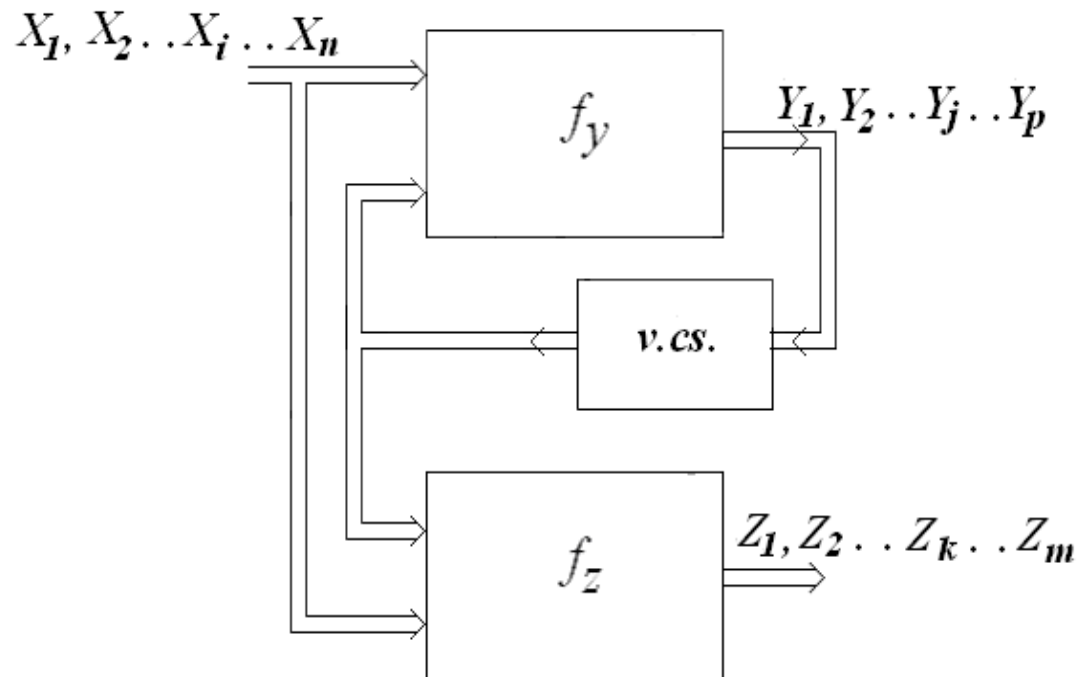
$y_l^{t+\Delta t}$  : egy szekundér változó kombináció  $(t + \Delta t)$  pillanatban

## Sorrendi hálózatok strukturális modelljei(1)

- Mealy-típusú szekvenciális hálózatok
  - Szinkron
  - Aszinkron
  
- Moore-típusú szekvenciális hálózatok
  - Szinkron
  - Aszinkron

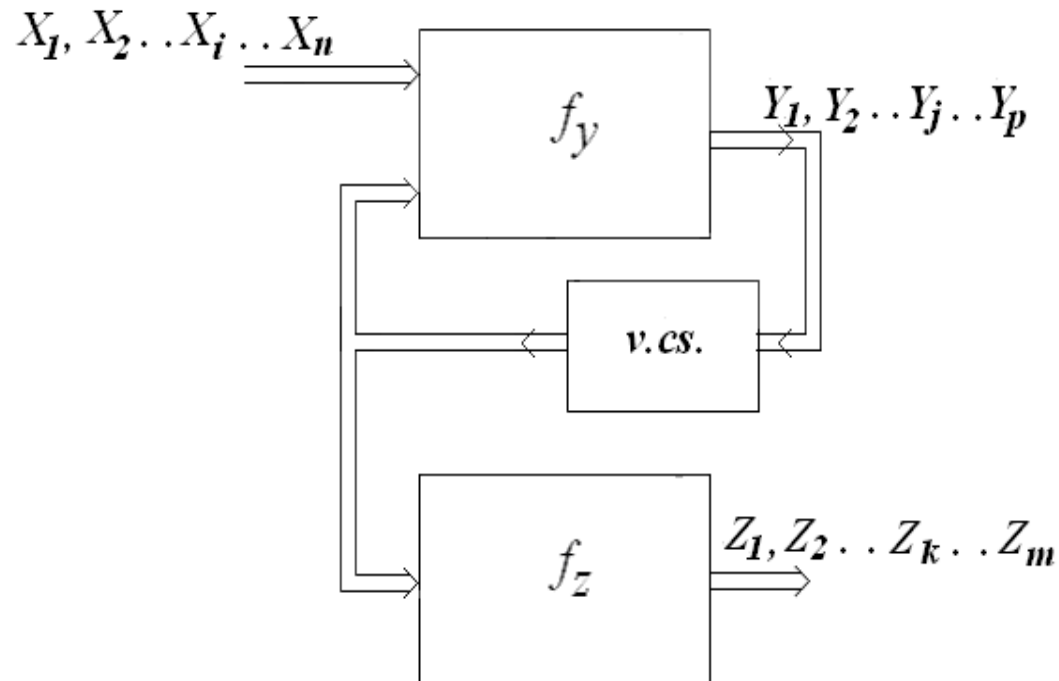
## Sorrendi hálózatok strukturális modelljei(2)

### Mealy-típusú szekvenciális hálózatok



## Sorrendi hálózatok strukturális modelljei(3)

### Moore-típusú szekvenciális hálózatok



## Sorrendi hálózatok csoportosítása

### ➤ Szinkron szekvenciális hálózatok

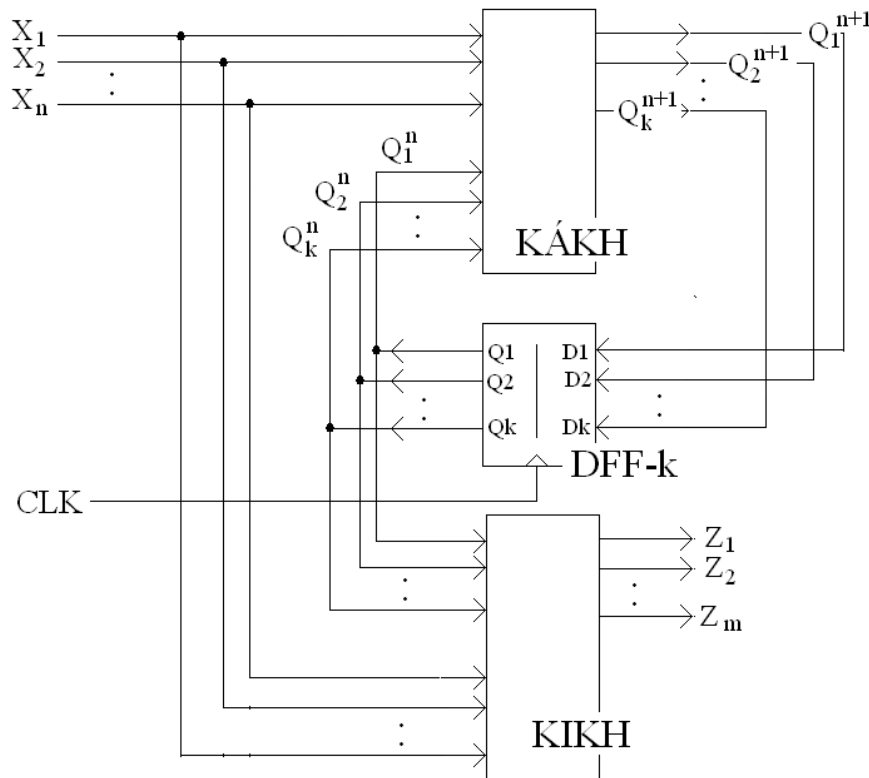
- *tárolóval („MS”) visszacsatolt szinkron sorrendi hálózatok*

### ➤ Aszinkron szekvenciális hálózatok

- *közvetlen visszacsatolásos aszinkron sorrendi hálózatok*
- *tárolóval visszacsatolt aszinkron sorrendi hálóztok*

## Sorrendi hálózatok felépítése és működése (1) (általános sémák)

**Példa: szinkron MEALY-hálózat, D-MS tárolós visszacsatoló ágakkal**

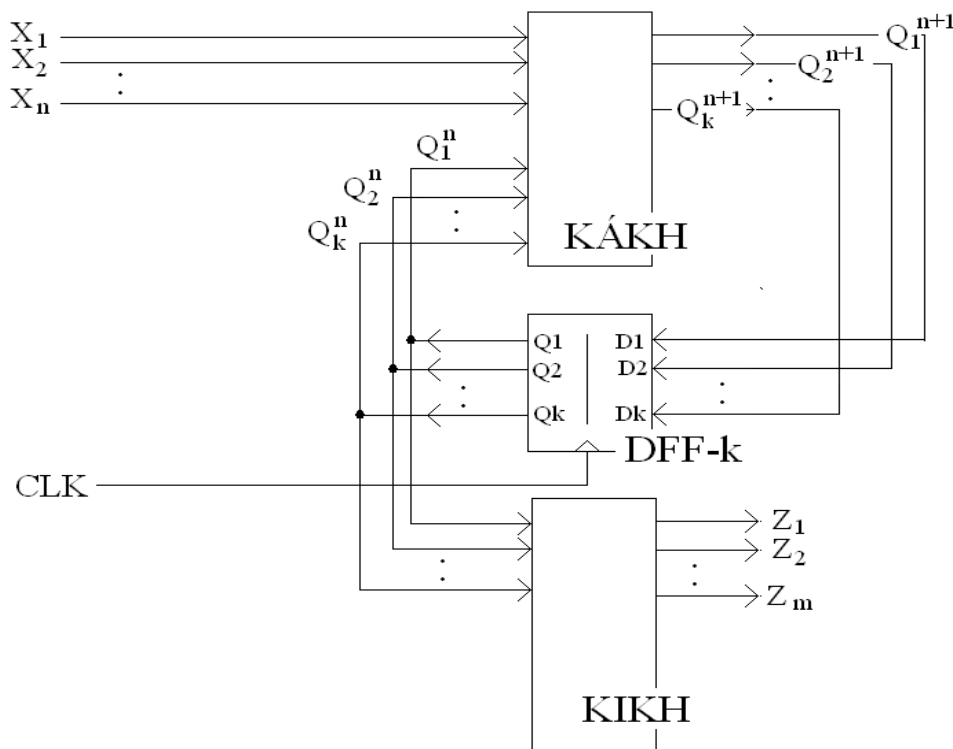


„KÁKH”: következő állapotot  
előállító kombinációs  
hálózat

„KIKH”: kimenetet előállító  
kombinációs hálózat

## Sorrendi hálózatok felépítése és működése (2) (általános sémák)

**Példa: szinkron MOORE-hálózat, D-MS tárolós visszacsatoló ágakkal**

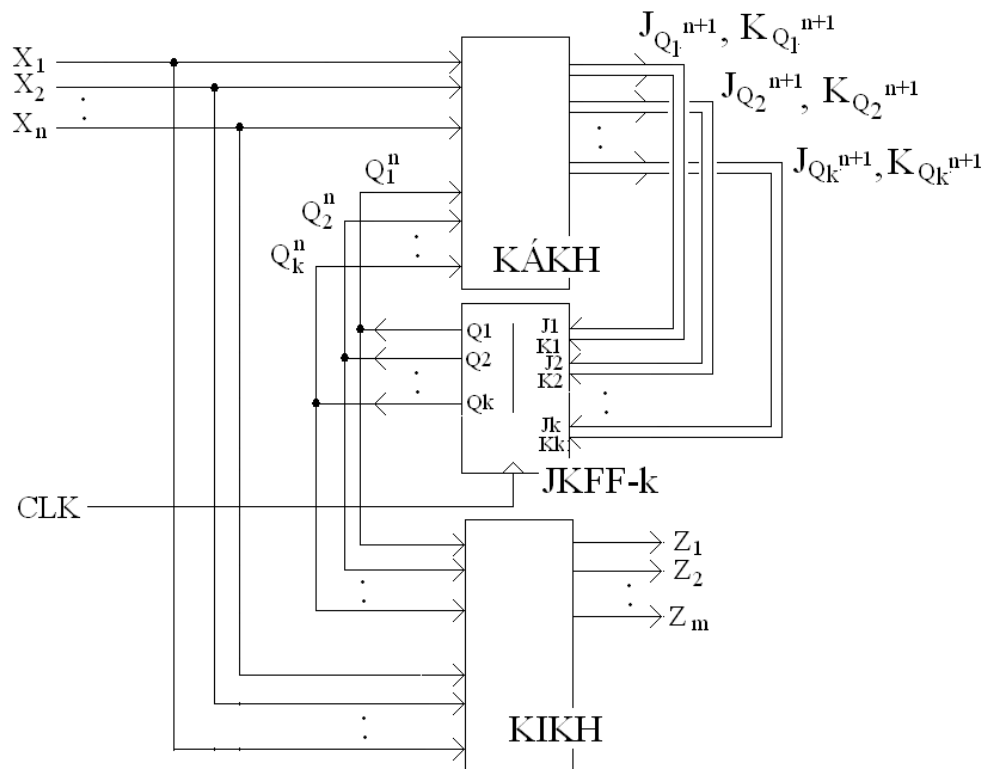


„KÁKH”: következő állapotot  
előállító kombinációs  
hálózat

„KIKH”: kimenetet előállító  
kombinációs hálózat

## Sorrendi hálózatok felépítése és működése (3) (általános sémák)

**Példa: szinkron MEALY-hálózat, JK-MS tárolós visszacsatoló ágakkal**

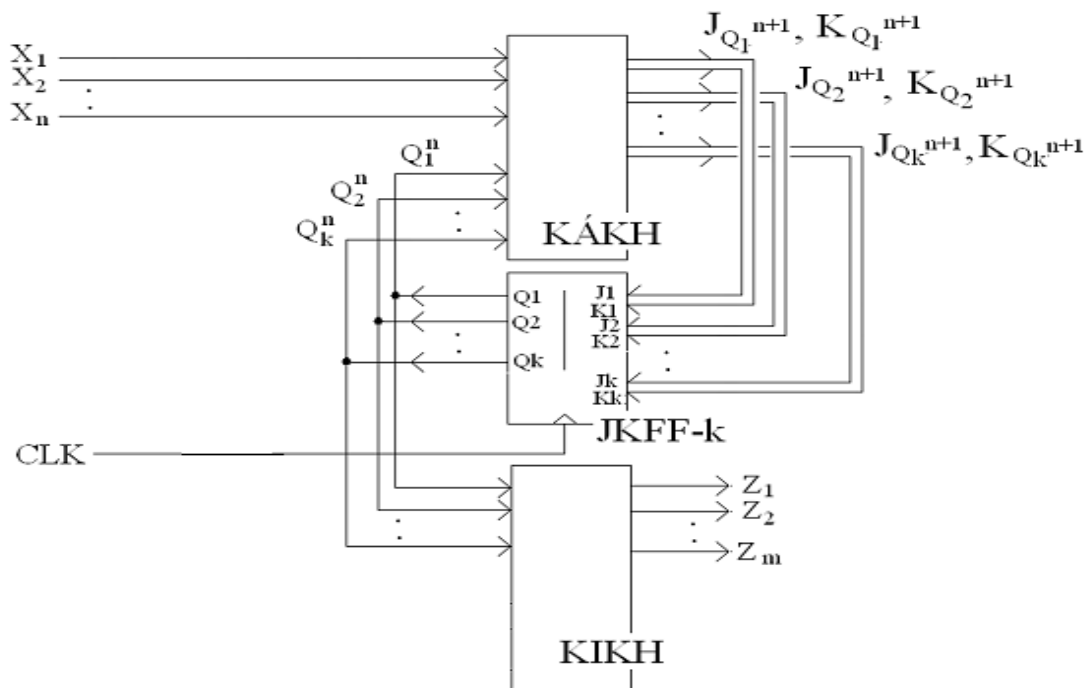


„KÁKH”: következő állapotot előállító kombinációs hálózat

„KIKH”: kimenetet előállító kombinációs hálózat

## Sorrendi hálózatok felépítése és működése (4) (általános sémák)

Példa: szinkron MOORE-hálózat, JK-MS tárolós visszacsatoló ágakkal

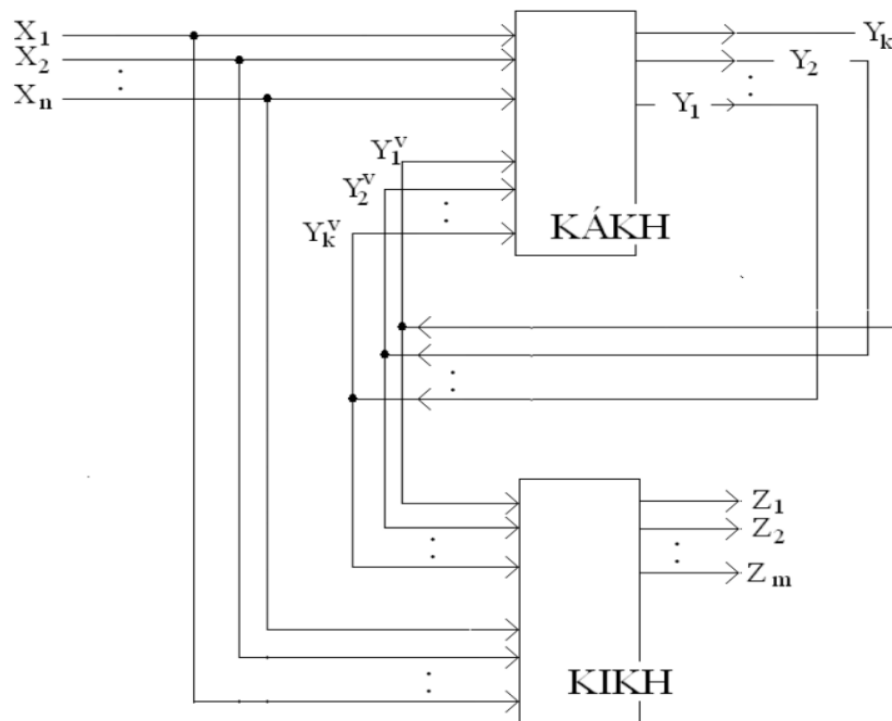


„KÁKH”: következő állapotot  
előállító kombinációs  
hálózat

„KIKH”: kimenetet előállító  
kombinációs hálózat

## Sorrendi hálózatok felépítése és működése (5) (általános sémák)

**Példa: aszinkron MEALY-hálózat, közvetlen visszacsatolásokkal**

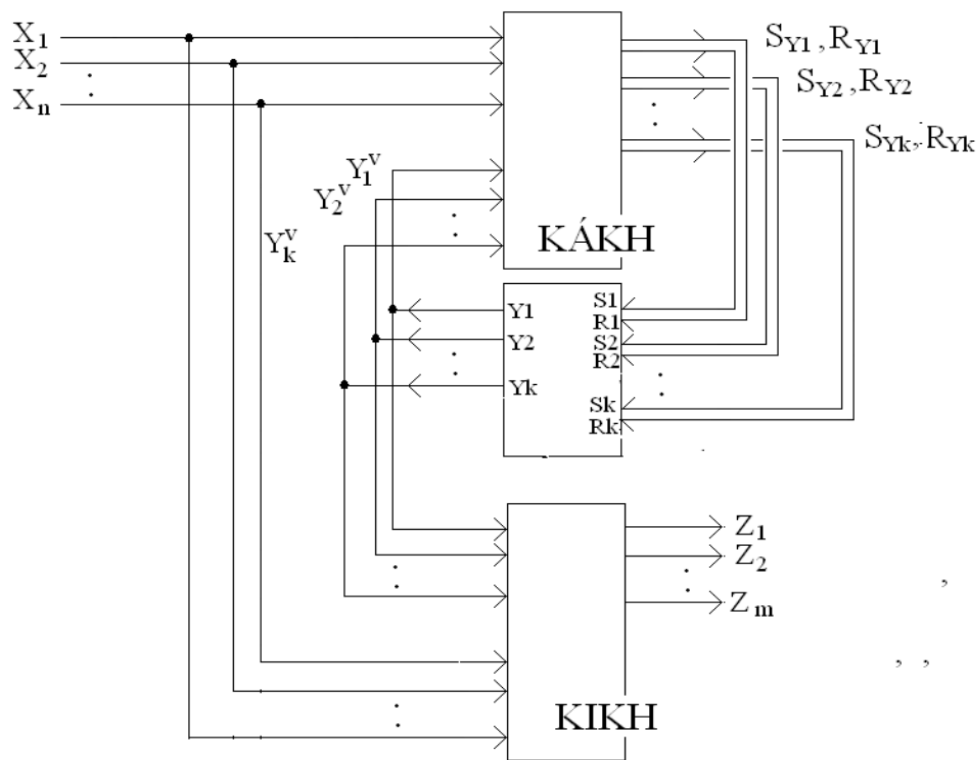


„KÁKH”: következő állapotot  
előállító kombinációs  
hálózat

„KIKH”: kimenetet előállító  
kombinációs hálózat

## Sorrendi hálózatok felépítése és működése (6) (általános sémák)

**Példa: aszinkron MEALY-hálózat, SR tárolós visszacsatolásokkal**

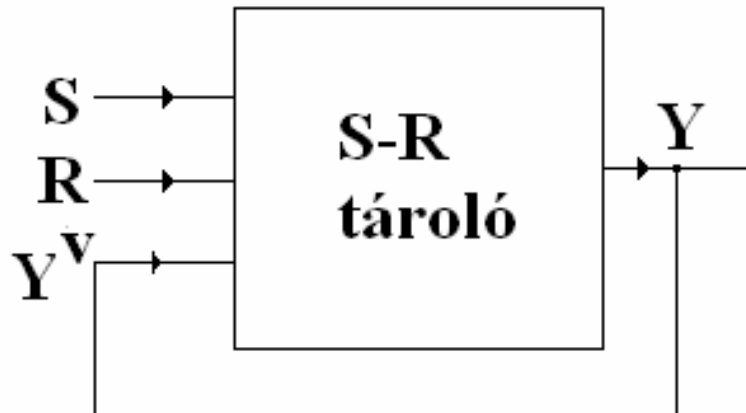


„KÁKH”: következő állapotot  
előállító kombinációs  
hálózat

„KIKH”: kimenetet előállító  
kombinációs hálózat

## Tárolók sorrendi hálózatokban: SR tároló (1)

Általános séma:



- kombinációs hálózat, amelynek kimenete a bemenetre érkezik vissza

## Tárolók sorrendi hálózatokban: SR tároló (2)

*egyszerű igazságtábla*

| S | R | Y     |
|---|---|-------|
| 0 | 0 | $Y^V$ |
| 0 | 1 | 0     |
| 1 | 0 | 1     |
| 1 | 1 | -     |



*összetett igazságtábla*

| S | R | $Y^V$ | Y |
|---|---|-------|---|
| 0 | 0 | 0     | 0 |
| 0 | 0 | 1     | 1 |
| 0 | 1 | 0     | 0 |
| 0 | 1 | 1     | 0 |
| 1 | 0 | 0     | 1 |
| 1 | 0 | 1     | 1 |
| 1 | 1 | 0     | - |
| 1 | 1 | 1     | - |

## Tárolók sorrendi hálózatokban: SR tároló (3)

Az állapot-átmeneti tábla

| $Y^v \backslash SR$ |  |      |      |      |      |
|---------------------|--|------|------|------|------|
|                     |  | $00$ | $01$ | $10$ | $11$ |
| $0$                 |  | $0$  | $0$  | $1$  | -    |
| $1$                 |  | $1$  | $0$  | $1$  | -    |

## Tárolók sorrendi hálózatokban: SR tároló (4)

*összetett igazságtábla*

| S | R | $Y^v$ | Y |
|---|---|-------|---|
| 0 | 0 | 0     | 0 |
| 0 | 0 | 1     | 1 |
| 0 | 1 | 0     | 0 |
| 0 | 1 | 1     | 0 |
| 1 | 0 | 0     | 1 |
| 1 | 0 | 1     | 1 |
| 1 | 1 | 0     | - |
| 1 | 1 | 1     | - |

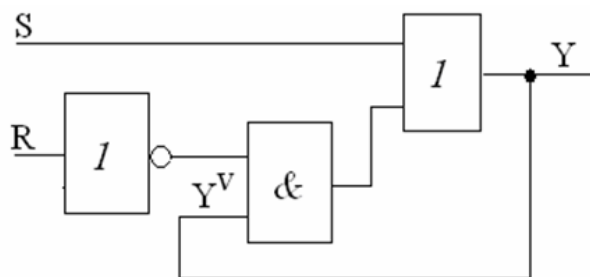
$\Rightarrow$

|       |   |   |   |   |   |
|-------|---|---|---|---|---|
|       | S | 0 | 0 | 1 | 1 |
|       | R | 0 | 1 | 1 | 0 |
| $Y^v$ | 0 | 0 | 0 | - | 1 |
| 1     | 1 | 0 | - | 1 |   |

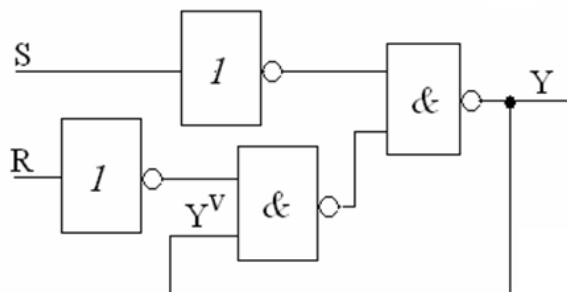
$$Y = S + \overline{R} Y^v = \overline{\overline{S} + \overline{R} Y^v} = \overline{\overline{S}(\overline{R} Y^v)}$$

## Tárolók sorrendi hálózatokban: SR tároló (5)

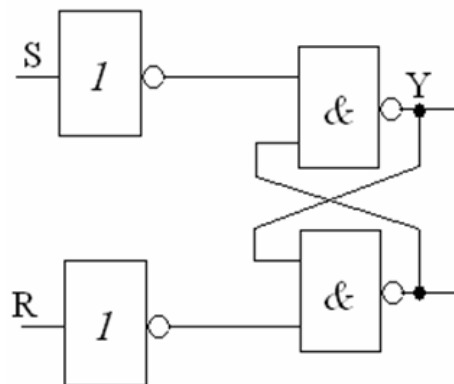
Realizációs megoldások:



• „ÉS-VAGY”

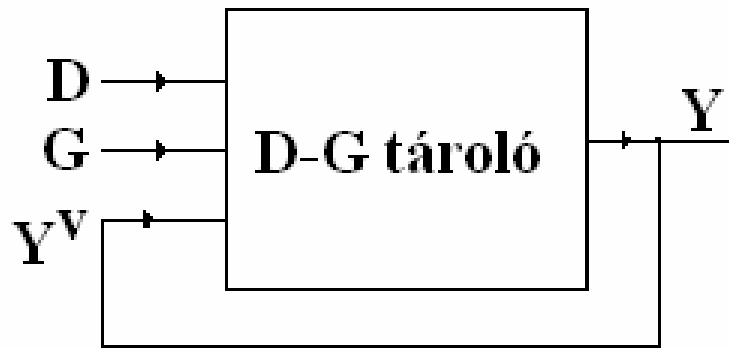


• „NEM-ÉS”



## Tárolók sorrendi hálózatokban: DG tároló (1)

Általános séma:



- kombinációs hálózat, amelynek kimenete a bemenetre érkezik vissza

## Tárolók sorrendi hálózatokban: DG tároló (2)

*egyszerű igazságtábla*

| D | G | Y     |
|---|---|-------|
| 0 | 0 | $Y^V$ |
| 0 | 1 | 0     |
| 1 | 0 | $Y^V$ |
| 1 | 1 | 1     |

$\Rightarrow$

*összetett igazságtábla*

| D | G | $Y^V$ | Y |
|---|---|-------|---|
| 0 | 0 | 0     | 0 |
| 0 | 0 | 1     | 1 |
| 0 | 1 | 0     | 0 |
| 0 | 1 | 1     | 0 |
| 1 | 0 | 0     | 0 |
| 1 | 0 | 1     | 1 |
| 1 | 1 | 0     | 1 |
| 1 | 1 | 1     | 1 |

## Tárolók sorrendi hálózatokban: DG tároló (3)

Az állapot-átmeneti tábla

| $Y^v$ \ $DG$ |      |      |      |      |
|--------------|------|------|------|------|
|              | $00$ | $01$ | $10$ | $11$ |
| $0$          | $0$  | $0$  | $0$  | $1$  |
| $1$          | $1$  | $0$  | $1$  | $1$  |

## Tárolók sorrendi hálózatokban: DG tároló (4)

*összetett igazságtábla*

| D | G | $Y^v$ | Y |
|---|---|-------|---|
| 0 | 0 | 0     | 0 |
| 0 | 0 | 1     | 1 |
| 0 | 1 | 0     | 0 |
| 0 | 1 | 1     | 0 |
| 1 | 0 | 0     | 0 |
| 1 | 0 | 1     | 1 |
| 1 | 1 | 0     | 1 |
| 1 | 1 | 1     | 1 |

$\Rightarrow$

|       | D | 0 | 0 | 1 | 1 |
|-------|---|---|---|---|---|
| G     | 0 | 1 | 1 | 0 |   |
| $Y^v$ |   |   |   |   |   |
| 0     | 0 | 0 | 1 | 0 |   |
| 1     | 1 | 0 | 1 | 1 |   |

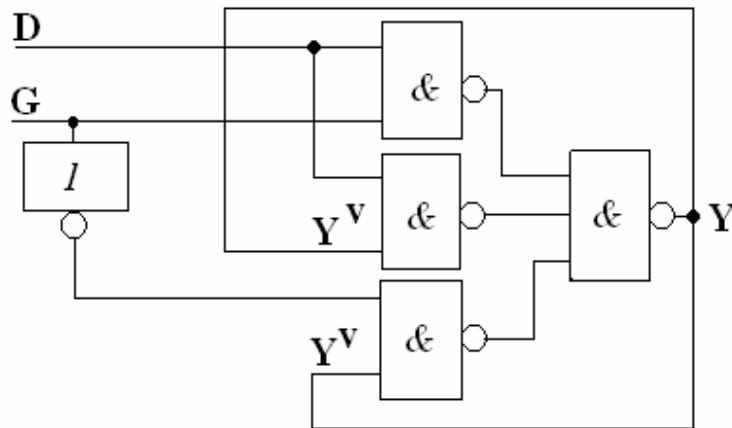
hazárdmentesítés!!

$$Y = D G + \underbrace{D Y^v}_{\text{hazárdmentesítés!!}} + \bar{G} Y^v$$

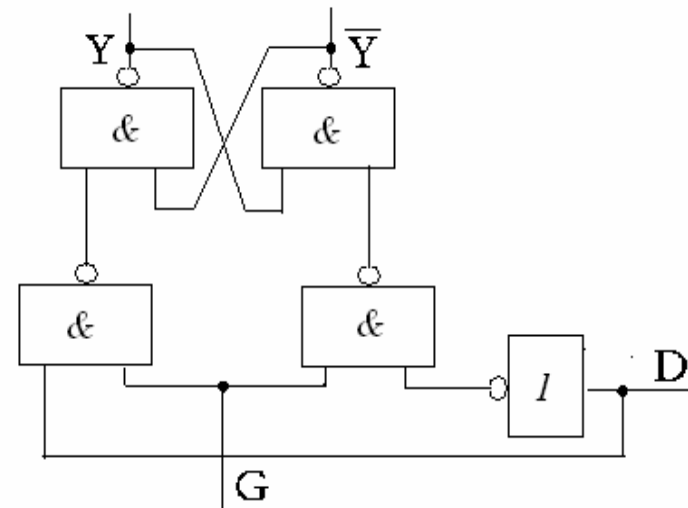
*Szabály: visszacsatolt kombinációs hálózattal  
megvalósított kapcsolást mindig hazárdmentesíteni kell !!!*

## Tárolók sorrendi hálózatokban: DG tároló (5)

Realizációs megoldások:



- „NAND-NAND”(1)



- „NAND-NAND”(2), SR-ből

## Tárolók sorrendi hálózatokban: **DG** tároló (6)

*A többszörös bemeneti szintváltás szemléltetése **DG** tárolón*

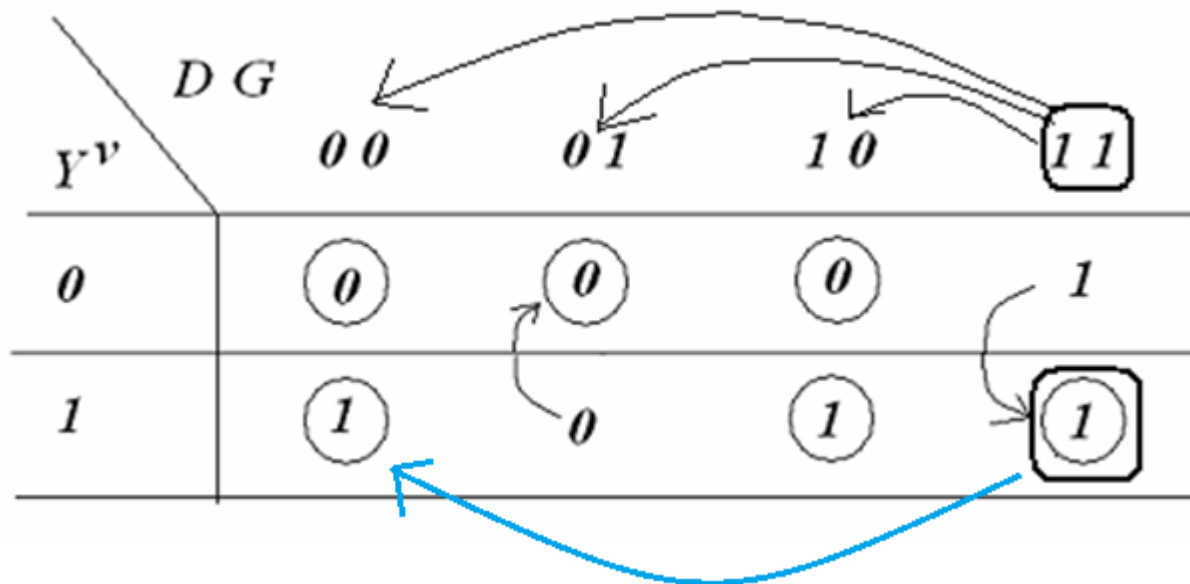
### **Példa:**

vizsgáljuk meg a következő bemeneti jelváltozást a **DG** tároló esetében az állapot-átmeneti tábla (állapottábla) segítségével!

DGY<sup>v</sup>: '111' → '001'

## Tárolók sorrendi hálózatokban: DG tároló (7)

*A többszörös bemeneti szintváltás szemléltetése **DG** tárolón*



DGY<sup>v</sup>: '111' → '001'

## Tárolók sorrendi hálózatokban: DG tároló (8)

*A többszörös bemeneti szintváltás szemléltetése **DG** tárolón*

A valóság:

DGY<sup>v</sup>: '111'  $\nrightarrow$  '001'



1. '111'  $\rightarrow$  '101'  $\rightarrow$  '001' vagy

2. '111'  $\rightarrow$  '011'  $\rightarrow$  '001' : ez az átmenet hibás működéshez vezet,  
mert Y=0-ban stabilizálódik a kimenet!!

## Tárolók sorrendi hálózatokban: DG tároló (9)

*A többszörös bemeneti szintváltás szemléltetése **DG** tárolón*

### Konklúzió:

A fenti lehetőségnek a kiküszöbölése csak akkor biztosítható, ha megtiltjuk a többszörös bemeneti váltásokat, azaz egyszerre csak egyetlen egy bemeneti jel értéke változhat meg. Ezzel egy olyan általános szabályt is kimondhatunk, mely szerint *visszacsatolt kombinációs hálózatok bemenetei közül egyszerre csak egyet szabad változtatni!*

## Tárolók sorrendi hálózatokban: DG tároló (10)

### *A DG tároló átlátszósága*

#### A probléma:

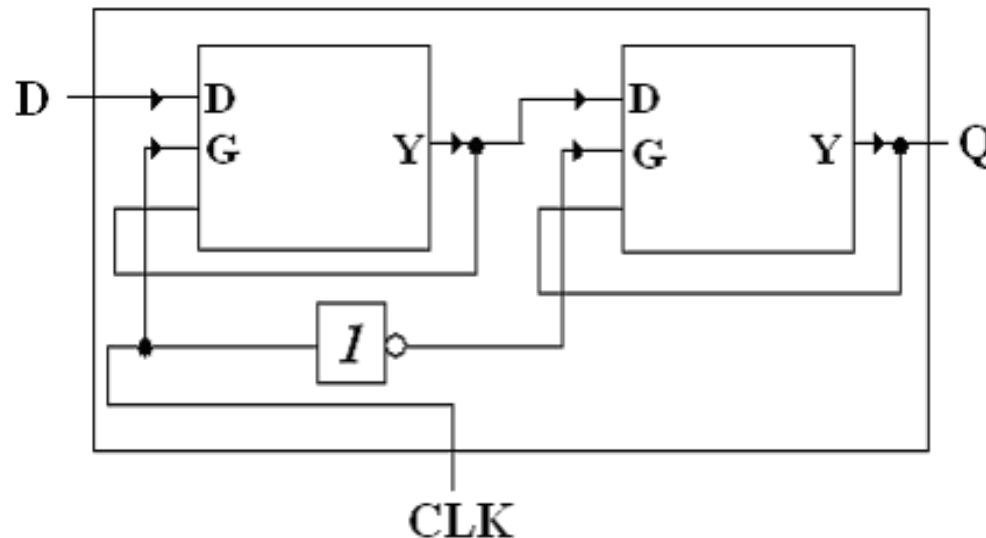
a **DG** tároló sajátossága, hogy a **G=1** helyzetben a **D**-re adott változások kijutnak a kimenetre. A **G=1** helyzetben tehát a tároló a **D**-bemenet felől „átlátszó” (transzparens).

#### Megoldás:

mester-szolga (master-slave) elv alkalmazása

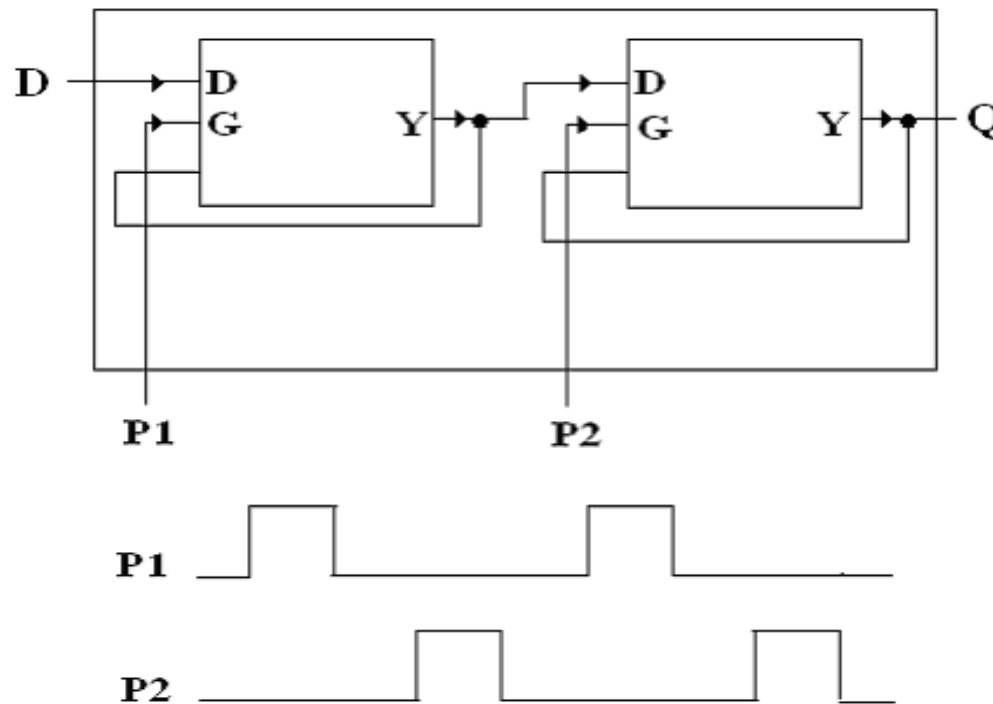
## Tárolók sorrendi hálózatokban: D-MS tároló(1)

*A **D-MS** tároló megvalósítása **DG** tárolókból a mester-szolga elv alkalmazásával: egyfázisú órajel alkalmazása*



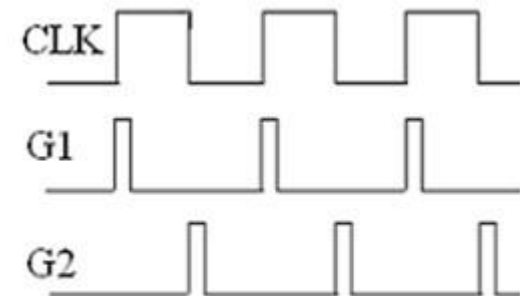
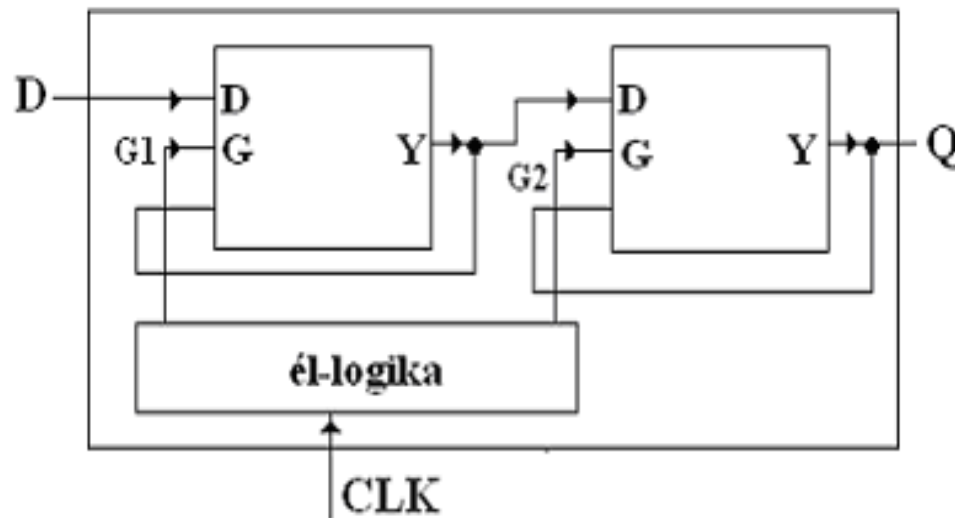
## Tárolók sorrendi hálózatokban: D-MS tároló(2)

A **D-MS** tároló megvalósítása **DG** tárolókból a mester-szolga elv alkalmazásával: kétfázisú, 'nem átlapolt' órajel alkalmazása



## Tárolók sorrendi hálózatokban: D-MS tároló(3)

*A **D-MS** tároló megvalósítása **DG** tárolókból a mester-szolga elv alkalmazásával: élvezérelt órajel alkalmazása*



## Tárolók sorrendi hálózatokban: D-MS tároló(4)

A **D-MS** tároló vezérlési táblája:

| D | $Q^n$ | $Q^{n+1}$ | $Q^n \rightarrow Q^{n+1}$ | D |
|---|-------|-----------|---------------------------|---|
| 0 | 0     | 0         | 0 0                       | 0 |
| 0 | 1     | 0         | 0 1                       | 1 |
| 1 | 0     | 1         | 1 0                       | 0 |
| 1 | 1     | 1         | 1 1                       | 1 |



| D | $Q^{n+1}$ |
|---|-----------|
| 0 | 0         |
| 1 | 1         |

## Tárolók sorrendi hálózatokban: JK-MS tároló(1)

*egyszerű igazságtábla*

| J | K | $Q^{n+1}$        |
|---|---|------------------|
| 0 | 0 | $Q^n$            |
| 0 | 1 | 0                |
| 1 | 0 | 1                |
| 1 | 1 | $\overline{Q^n}$ |

$\Rightarrow$

*összetett igazságtábla*

| J | K | $Q^n$ | $Q^{n+1}$ |
|---|---|-------|-----------|
| 0 | 0 | 0     | 0         |
| 0 | 0 | 1     | 1         |
| 0 | 1 | 0     | 0         |
| 0 | 1 | 1     | 0         |
| 1 | 0 | 0     | 1         |
| 1 | 0 | 1     | 1         |
| 1 | 1 | 0     | 1         |
| 1 | 1 | 1     | 0         |

## Tárolók sorrendi hálózatokban: JK-MS tároló(2)

*A JK-MS flip-flop megvalósítása D tárolóval: egy gyakran használt megoldás*

*összetett igazságtábla*

| J | K | $Q^n$ | $Q^{n+1}$ |
|---|---|-------|-----------|
| 0 | 0 | 0     | 0         |
| 0 | 0 | 1     | 1         |
| 0 | 1 | 0     | 0         |
| 0 | 1 | 1     | 0         |
| 1 | 0 | 0     | 1         |
| 1 | 0 | 1     | 1         |
| 1 | 1 | 0     | 1         |
| 1 | 1 | 1     | 0         |

$\Rightarrow$

*A D bemenet vezérlése*

| J | K | $Q^n$ | D |
|---|---|-------|---|
| 0 | 0 | 0     | 0 |
| 0 | 0 | 1     | 1 |
| 0 | 1 | 0     | 0 |
| 0 | 1 | 1     | 0 |
| 1 | 0 | 0     | 1 |
| 1 | 0 | 1     | 1 |
| 1 | 1 | 0     | 1 |
| 1 | 1 | 1     | 0 |

## Tárolók sorrendi hálózatokban: JK-MS tároló(3)

*A JK-MS flip-flop megvalósítása D tárolóval: egy gyakran használt megoldás*

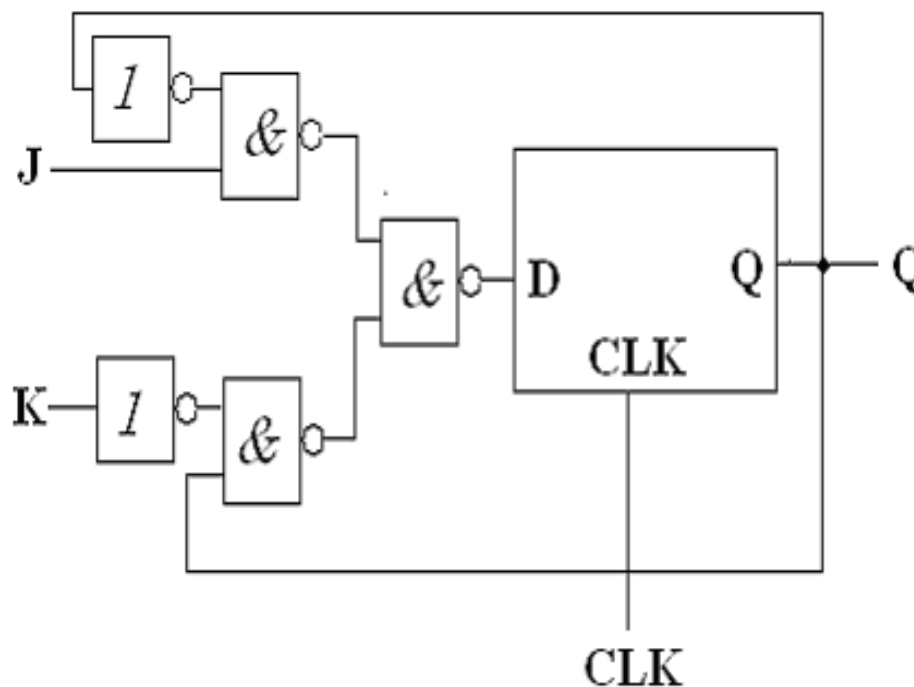
| D              | J | 0 | 0 | 1 | 1 |
|----------------|---|---|---|---|---|
|                | K | 0 | 1 | 1 | 0 |
| Q <sup>n</sup> | 0 |   |   | 1 | 1 |
|                | 1 | 1 |   |   | 1 |

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

$$(D = \overline{\overline{J\overline{Q}^n}} \cdot \overline{\overline{\overline{K}Q^n}})$$

## Tárolók sorrendi hálózatokban: JK-MS tároló(4)

Realizáció:



$$(D = \overline{\overline{J\overline{Q^n}}}) \cdot \overline{\overline{KQ^n}})$$

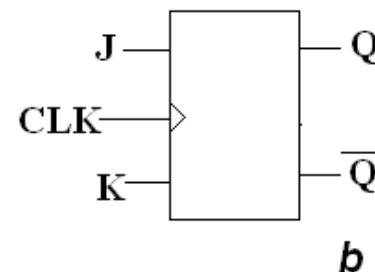
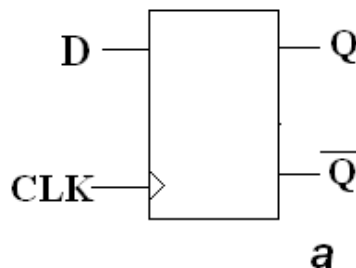
## Tárolók sorrendi hálózatokban: JK-MS tároló(5)

*A JK-MS flip-flop vezérlési táblája:*

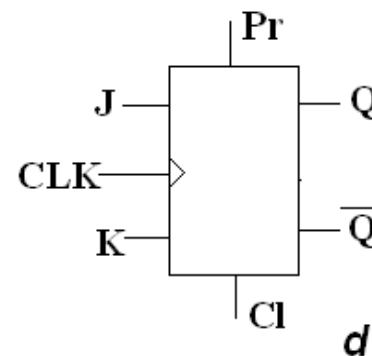
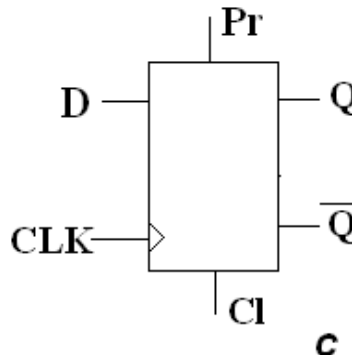
| J | K | $Q^n$ | $Q^{n+1}$ | $Q^n \rightarrow Q^{n+1}$ | J | K |
|---|---|-------|-----------|---------------------------|---|---|
| 0 | 0 | 0     | 0         | 0 → 0                     | 0 | — |
| 0 | 0 | 1     | 1         | 0 → 1                     | 1 | — |
| 0 | 1 | 0     | 0         | 1 → 0                     | — | 1 |
| 0 | 1 | 1     | 0         | 1 → 1                     | — | 0 |
| 1 | 0 | 0     | 1         |                           |   |   |
| 1 | 0 | 1     | 1         |                           |   |   |
| 1 | 1 | 0     | 1         |                           |   |   |
| 1 | 1 | 1     | 0         |                           |   |   |

## Flip-flopok segéd-bemenetei és szimbólumaik

Segédbemenetek nélkül:



Segédbemenetekkel:



*Pr (Preset) : az aktuális állapottól függetlenül ,1'-be állítja a tárolót(Q)*

*Cl (Clear) : az aktuális állapottól függetlenül ,0'-ba állítja a tárolót(Q)*

## Szinkron sorrendi hálózatok tervezése mintapéldákon keresztül I.

### - 1. mintafeladat -

*Adott egy kétbemenetű ( $X_1, X_2$ ) és egy kimenettel ( $Z$ ) rendelkező szinkron sorrendi hálózat. A hálózat az első  $X_1=X_2$  bemeneti kombinációtól kezdve vizsgálja a bemeneteket, és a  $Z$  kimenetén jelzi, ha a két bemenet kétszer egymás után azonos logikai szintű. Ha ilyen kombináció sorozat lezajlott, a vizsgálatot újra kezdi. Tervezzük meg a hálózatot **JK-MS** tárolókkal!*

## - 1. mintafeladat -

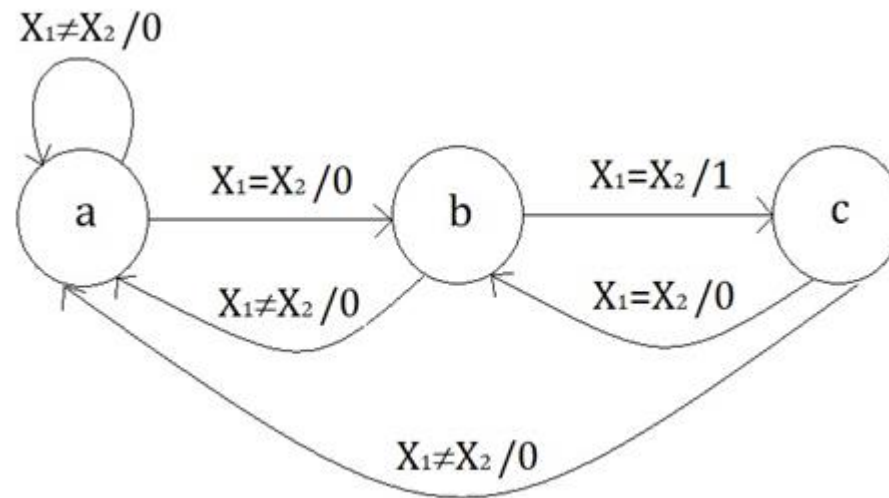
*A feladat megoldásához a következő, általános szisztematikus lépéssort alkalmazzuk:*

- 1. Állapot-átmeneti gráf felrajzolása.*
- 2. Előzetes szimbolikus állapottábla felvétele.*
- 3. Összevont szimbolikus állapottábla megszerkesztése.*
- 4. Kódolt állapottábla elkészítése.*
- 5. A specifikációra érvényes vezérlési tábla elkészítése (tárolók használata esetében).*
- 6. A szekunder változó(k) és a kimenet(ek) függvényeinek Karnaugh tábla segítségével történő megadása.*
- 7. Kezdeti állapotról történő gondoskodás.*
- 8. Realizáció logikai kapukkal (és tárolókkal).*

## - 1. mintafeladat -

### Megoldás 1.(Mealy-modell):

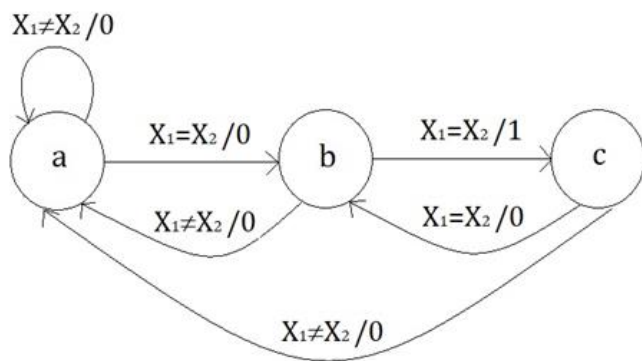
#### 1. Az állapot-átmeneti gráf (állapotgráf) felrajzolása (Mealy):



## - 1. mintafeladat -

### 2. Az előzetes, szimbolikus állapottábla felvétele (Mealy):

*előzetes szimbolikus állapottábla*



| aktuális állapot | következő állapot/kimenet |             |
|------------------|---------------------------|-------------|
|                  | $X_1 \neq X_2$            | $X_1 = X_2$ |
| $a$              | $a/0$                     | $b/0$       |
| $b$              | $a/0$                     | $c/1$       |
| $c$              | $a/0$                     | $b/0$       |

## - 1. mintafeladat -

### 3. Összevont szimbolikus állapottábla megszerkesztése (Mealy):

Az állapot összevonás feltételei:

***az előzetes szimbolikus állapottábla két állapotát nem kell megkülönböztetni, ezért azok összevonhatók,***

- ***ha bemeneti kombinációként egyeznek a hozzájuk rendelt kimeneti kombinációk,***
- ***és bemenő kombinációként ugyanarra a következő állapotra vezetnek.***

Példánkban az ,a' és a ,c' állapotok összevonhatók (,ac' ; ,b')

## - 1. mintafeladat -

- A bemeneti egyszerűsítési lehetőségek kihasználása!

Vegyük észre! ☺ : a két bemenet helyett csak egy bemenetet kell figyelnünk a feladat megoldása során!

$$E = X_1 X_2 + \overline{X_1} \overline{X_2}$$

↑

$$\{X_1 = X_2 (00; 11) : \text{exnor} = ,1'\}$$



összevont szimbolikus állapottábla

| aktuális állapot | következő állapot/kimenet |        |
|------------------|---------------------------|--------|
|                  | $\bar{E}$                 | $E$    |
| $ac$             | $ac/0$                    | $b/0$  |
| $b$              | $ac/0$                    | $ac/1$ |

## - 1. mintafeladat -

### 4. Kódolt állapottábla elkészítése (Mealy):

- másodlagos (szekunder) ,belső' változó (Q) bevezetése az állapotok megkülönböztetésére:

$$ac \rightarrow Q=0$$

$$b \rightarrow Q=1$$



*kódolt állapottábla*

| kódolt aktuális<br>állapot,<br>$Q^n$ | kódolt következő állapot/kimenet |                    |
|--------------------------------------|----------------------------------|--------------------|
|                                      | $\bar{E}$<br>$Q^{n+1}/Z$         | $E$<br>$Q^{n+1}/Z$ |
| 0                                    | 0/0                              | 1/0                |
| 1                                    | 0/0                              | 0/1                |

## - 1. mintafeladat -

### 5. A specifikációra érvényes vezérlési tábla elkészítése (Mealy):

A JK-MS saját  
vezérlési táblája

| $Q^n \rightarrow Q^{n+1}$ | J | K |
|---------------------------|---|---|
| $0 \rightarrow 0$         | 0 | - |
| $0 \rightarrow 1$         | 1 | - |
| $1 \rightarrow 0$         | - | 1 |
| $1 \rightarrow 1$         | - | 0 |

kódolt állapottábla

| kódolt<br>aktuális<br>állapot,<br>$Q^n$ | kódolt következő állapot/kimenet |                  |
|-----------------------------------------|----------------------------------|------------------|
|                                         | $\bar{E}$<br>$Q^{n+1}$           | $E$<br>$Q^{n+1}$ |
| 0                                       | 0/0                              | 1/0              |
| 1                                       | 0/0                              | 0/1              |

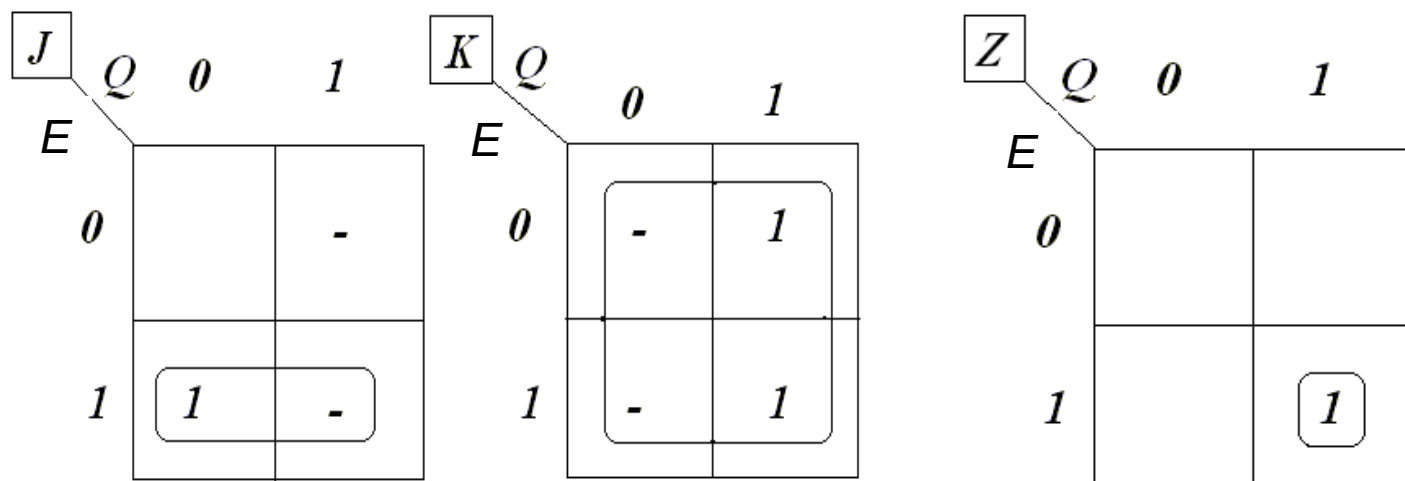


vezérlési tábla

| $Q$ | $\bar{E}$ |   | $E$ |   |
|-----|-----------|---|-----|---|
|     | J         | K | J   | K |
| 0   | 0         | - | 1   | - |
| 1   | -         | 1 | -   | 1 |

## - 1. mintafeladat -

6. A szekunder változó és a kimenet függvényeinek Karnaugh tábla segítségével történő megadása (Mealy):

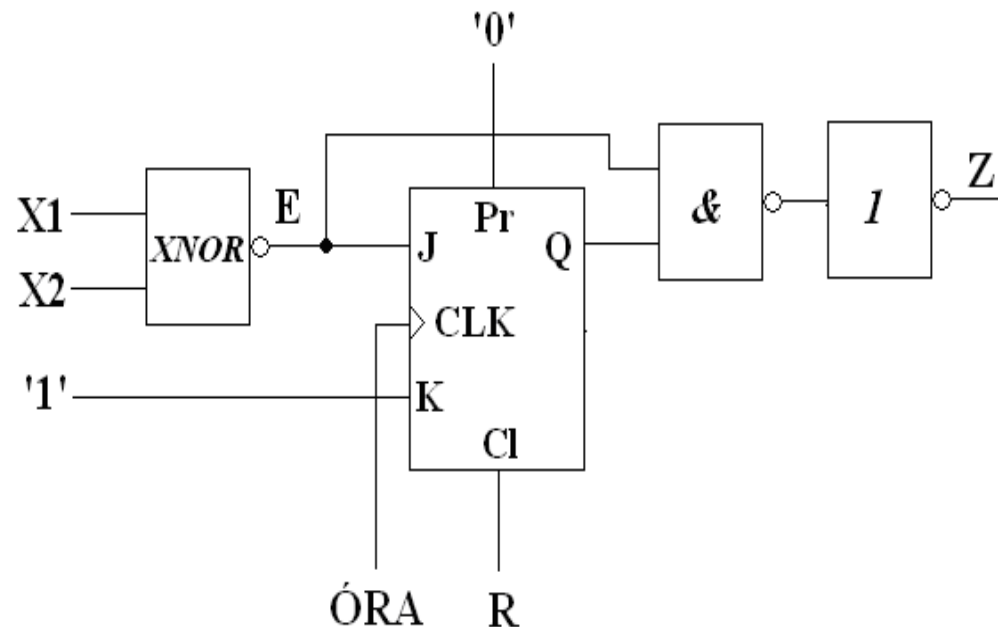


$$J = E, \quad K = 1, \quad Z = Q E$$

## - 1. mintafeladat -

7-8. Realizáció (,NAND'), a kezdeti állapot biztosítása (RESET) (Mealy):

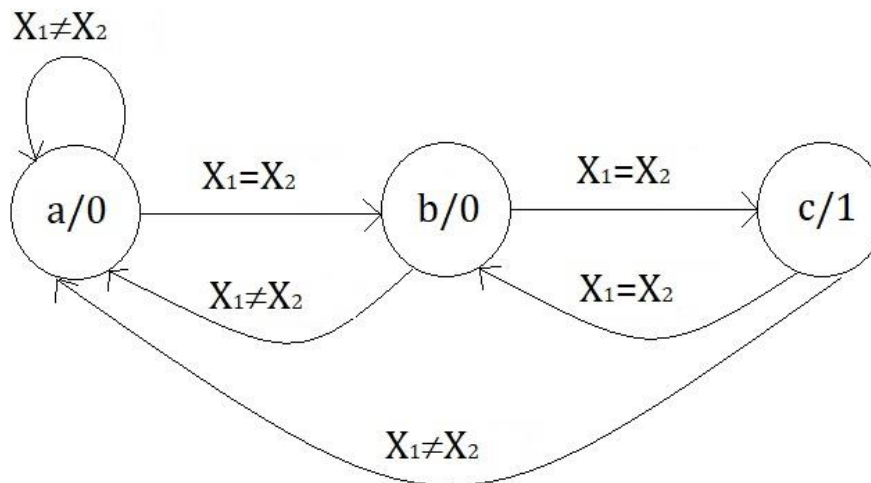
$$J = E, \quad K = 1, \quad Z = Q E \quad (,NAND' \rightarrow) \quad J = E, \quad K = 1, \quad Z = \overline{\overline{Q} E}$$



## - 1. mintafeladat -

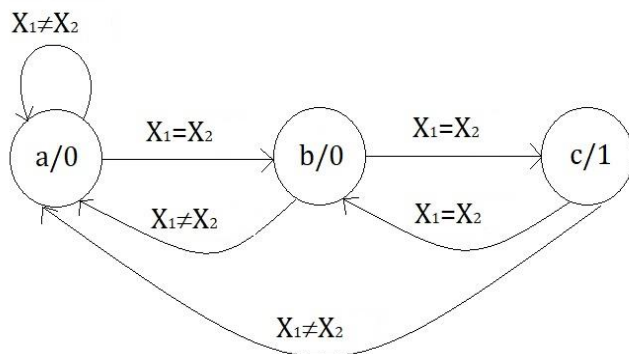
### Megoldás 2. (Moore-modell):

#### 1. Az állapotgráf felrajzolása (Moore):



## - 1. mintafeladat -

## 2. Az előzetes, szimbolikus állapottábla felvétele (Moore) :



*előzetes szimbolikus állapottábla*

| aktuális<br>állapot/kimenet | következő állapot |             |
|-----------------------------|-------------------|-------------|
|                             | $X_1 \neq X_2$    | $X_1 = X_2$ |
| $a/0$                       | $a$               | $b$         |
| $b/0$                       | $a$               | $c$         |
| $c/1$                       | $a$               | $b$         |

## - 1. mintafeladat -

### 3. Összevont szimbolikus állapottábla megszerkesztése (Moore):

*előzetes szimbolikus állapottábla*

| aktuális<br>állapot/kimenet | következő állapot |             |
|-----------------------------|-------------------|-------------|
|                             | $X_1 \neq X_2$    | $X_1 = X_2$ |
| $a/0$                       | $a$               | $b$         |
| $b/0$                       | $a$               | $c$         |
| $c/1$                       | $a$               | $b$         |

*Az előzetes szimbolikus állapottáblán – az összevonás feltételeit figyelembe véve- ezúttal nem tudunk állapot összevonást végezni. A kódolt állapottáblát így e táblázat alapján kell elkészíteni.*

## - 1. mintafeladat -

### 4. Kódolt állapottábla elkészítése (Moore):

- Szekunder változók bevezetése az állapotok megkülönböztetésére:

*kódolt állapottábla*

egy lehetséges kód kiosztás:

$$a \rightarrow Q_1 Q_2 = 00$$

$$b \rightarrow Q_1 Q_2 = 01 \Rightarrow$$

$$c \rightarrow Q_1 Q_2 = 10$$

| kódolt aktuális<br>állapot/kimenet | kódolt következő állapot |                       |
|------------------------------------|--------------------------|-----------------------|
|                                    | $\bar{E}$                | $E$                   |
| $Q_1^n Q_2^n / Z$                  | $Q_1^{n+1} Q_2^{n+1}$    | $Q_1^{n+1} Q_2^{n+1}$ |
| 0 0 / 0                            | 0 0                      | 0 1                   |
| 0 1 / 0                            | 0 0                      | 1 0                   |
| 1 0 / 1                            | 0 0                      | 0 1                   |
| 1 1 / -                            | - -                      | - -                   |

- 1. mintafeladat -

## 5. A specifikációra érvényes vezérlési tábla elkészítése (Moore):

A JK-MS saját  
vezérlési táblája

| $Q^n \rightarrow Q^{n+1}$ | J | K |
|---------------------------|---|---|
| $0 \rightarrow 0$         | 0 | - |
| $0 \rightarrow 1$         | 1 | - |
| $1 \rightarrow 0$         | - | 1 |
| $1 \rightarrow 1$         | - | 0 |

kódolt állapottábla

| kódolt aktuális<br>állapot/kimenet | kódolt következő állapot |                       |
|------------------------------------|--------------------------|-----------------------|
|                                    | $\bar{E}$                | $E$                   |
| $Q_1^n Q_2^n / Z$                  | $Q_1^{n+1} Q_2^{n+1}$    | $Q_1^{n+1} Q_2^{n+1}$ |
| $0\ 0 / 0$                         | $0\ 0$                   | $0\ 1$                |
| $0\ 1 / 0$                         | $0\ 0$                   | $1\ 0$                |
| $1\ 0 / 1$                         | $0\ 0$                   | $0\ 1$                |
| $1\ 1 / -$                         | $--$                     | $--$                  |



vezérlési tábla

| kódolt aktuális<br>állapot/kimenet | kódolt következő állapot |           |           |           |
|------------------------------------|--------------------------|-----------|-----------|-----------|
|                                    | $\bar{E}$                |           | $E$       |           |
|                                    | $J_1 K_1$                | $J_2 K_2$ | $J_1 K_1$ | $J_2 K_2$ |
| $Q_1 Q_2 / Z$                      |                          |           |           |           |
| $0\ 0 / 0$                         | $0 -$                    | $0 -$     | $0 -$     | $1 -$     |
| $0\ 1 / 0$                         | $0 -$                    | $- 1$     | $1 -$     | $- 1$     |
| $1\ 0 / 1$                         | $- 1$                    | $0 -$     | $- 1$     | $1 -$     |
| $1\ 1 / -$                         | $--$                     | $--$      | $--$      | $--$      |

## - 1. mintafeladat -

6. A szekunder változók és a kimenet függvényeinek Karnaugh tábla segítségével történő megadása (Moore):

**J1**

|    |   |   |   |   |
|----|---|---|---|---|
| Q1 | 0 | 0 | 1 | 1 |
| Q2 | 0 | 1 | 1 | 0 |
| E  |   |   |   |   |
| 0  |   |   | - | - |
| 1  |   | 1 | - | - |

$$J1 = Q2 \cdot E$$

**J2**

|    |   |   |   |   |
|----|---|---|---|---|
| Q1 | 0 | 0 | 1 | 1 |
| Q2 | 0 | 1 | 1 | 0 |
| E  |   |   |   |   |
| 0  |   | - | - |   |
| 1  | 1 | - | - | 1 |

$$J2 = E$$

**K1**

|    |   |   |   |   |
|----|---|---|---|---|
| Q1 | 0 | 0 | 1 | 1 |
| Q2 | 0 | 1 | 1 | 0 |
| E  |   |   |   |   |
| 0  | - | - | - | 1 |
| 1  | - | - | - | 1 |

$$K1 = 1$$

**K2**

|    |   |   |   |   |
|----|---|---|---|---|
| Q1 | 0 | 0 | 1 | 1 |
| Q2 | 0 | 1 | 1 | 0 |
| E  |   |   |   |   |
| 0  | - | 1 | - | - |
| 1  | - | 1 | - | - |

$$K2 = 1$$

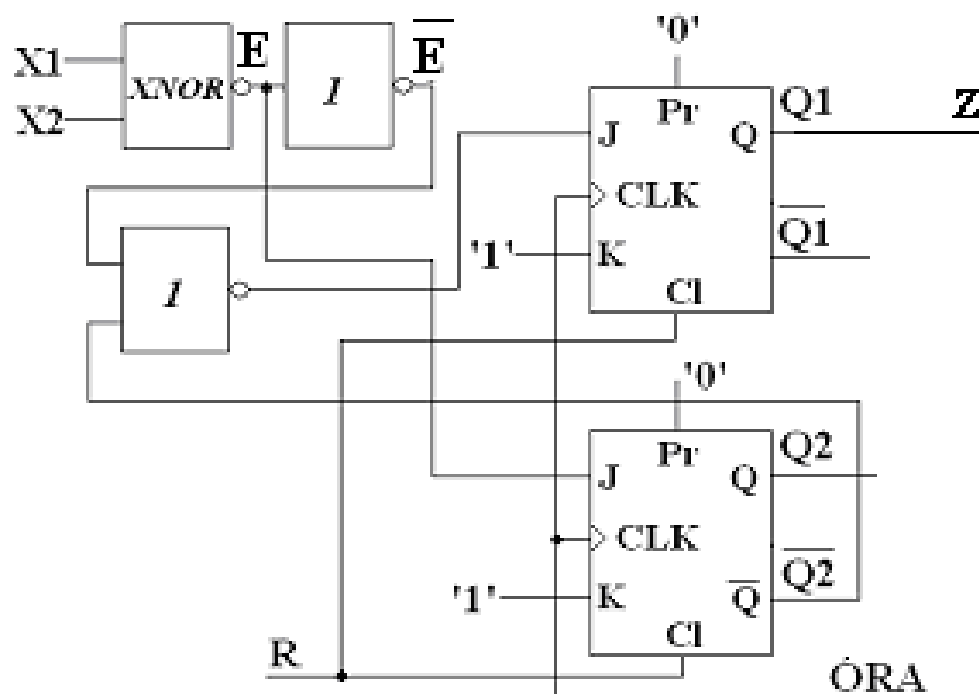
**Z**

|    |   |   |
|----|---|---|
| Q1 | 0 | 1 |
| Q2 | 0 | 1 |
| 0  |   | 1 |
| 1  |   | - |

$$Z = Q1$$

## - 1. mintafeladat -

7-8. Realizáció a kezdeti állapot biztosításával (RESET) (Moore):



## Szinkron sorrendi hálózatok tervezése mintapéldákon keresztül I.

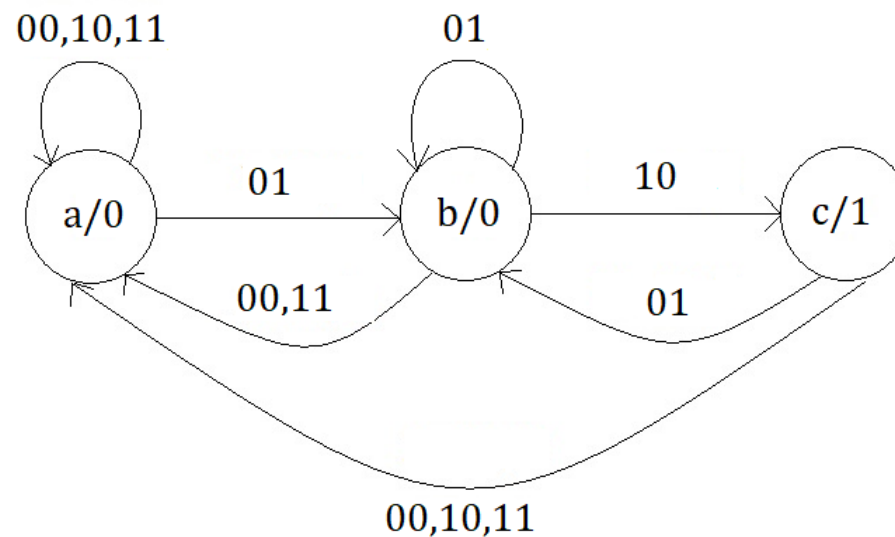
### - 2. mintafeladat -

*Tervezzük meg a lehető legegyszerűbb változatban azt a 'preset' és 'clear' bemenetekkel is rendelkező, élvezérelt **D-MS** tárolókkal felépített Moore-típusú szinkron sorrendi hálózatot, amelynek két bemenete ( $X_1, X_2$ ) és egy kimenete ( $Z$ ) van. A hálózat az órajel felfutása előtt fogadja a bemeneti kombinációkat. A hálózat  $Z$  kimenete akkor lesz '1', ha a bemenetre az '1 0' kombináció érkezik, de csak abban az esetben, ha az előző óraciklus által fogadott bemeneti bitkombináció '0 1' volt! A kezdeti állapothoz az  $X_1X_2 = '00'$  kombináció tartozik.*

## - 2. mintafeladat -

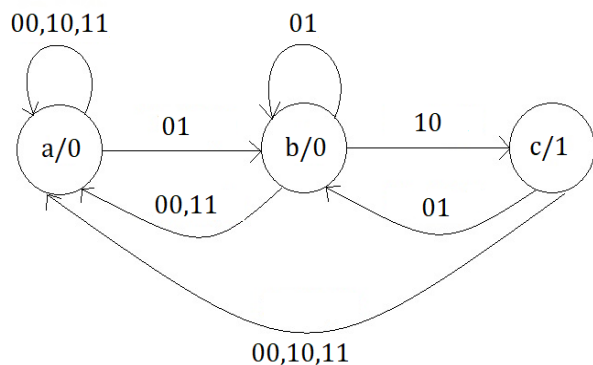
### Megoldás 1.(Moore-modell):

#### 1. Az állapotgráf felrajzolása (Moore):



## - 2. mintafeladat -

### 2. Az előzetes, szimbolikus állapottábla felvétele (Moore):



*előzetes szimbolikus állapottábla*

| aktuális<br>állapot/<br>kimenet | következő állapot |           |           |           |
|---------------------------------|-------------------|-----------|-----------|-----------|
|                                 | $X_1X_2$          |           |           |           |
|                                 | <i>00</i>         | <i>01</i> | <i>10</i> | <i>11</i> |
| <i>a/0</i>                      | <i>a</i>          | <i>b</i>  | <i>a</i>  | <i>a</i>  |
| <i>b/0</i>                      | <i>a</i>          | <i>b</i>  | <i>c</i>  | <i>a</i>  |
| <i>c/1</i>                      | <i>a</i>          | <i>b</i>  | <i>a</i>  | <i>a</i>  |

## - 2. mintafeladat -

### 3. Összevont szimbolikus állapottábla megszerkesztése (Moore):

*előzetes szimbolikus állapottábla*

| aktuális<br>állapot/<br>kimenet | következő állapot |      |      |      |
|---------------------------------|-------------------|------|------|------|
|                                 | $X_1X_2$          |      |      |      |
|                                 | $00$              | $01$ | $10$ | $11$ |
| $a/0$                           | $a$               | $b$  | $a$  | $a$  |
| $b/0$                           | $a$               | $b$  | $c$  | $a$  |
| $c/1$                           | $a$               | $b$  | $a$  | $a$  |

*Az előzetes szimbolikus állapottáblán – az összevonás feltételeit figyelembe véve – nem tudunk állapot összevonást végezni. A kódolt állapottáblát így e táblázat alapján kell elkészíteni.*

## - 2. mintafeladat -

### 4. Kódolt állapottábla elkészítése (Moore):

- Szekunder változók bevezetése az állapotok megkülönböztetésére:

kódolt állapottábla

egy lehetséges kódkiosztás:

$$a \rightarrow Q_1 Q_2 = 00$$

$$b \rightarrow Q_1 Q_2 = 01 \Rightarrow$$

$$c \rightarrow Q_1 Q_2 = 10$$

| kódolt aktuális<br>állapot/kimenet | kódolt következő állapot |                       |                       |                       |
|------------------------------------|--------------------------|-----------------------|-----------------------|-----------------------|
|                                    | $X_1 X_2$                |                       |                       |                       |
|                                    | $00$                     | $01$                  | $10$                  | $11$                  |
| $Q_1^n Q_2^n / Z$                  | $Q_1^{n+1} Q_2^{n+1}$    | $Q_1^{n+1} Q_2^{n+1}$ | $Q_1^{n+1} Q_2^{n+1}$ | $Q_1^{n+1} Q_2^{n+1}$ |
| $0\ 0/0$                           | $0\ 0$                   | $0\ 1$                | $0\ 0$                | $0\ 0$                |
| $0\ 1/0$                           | $0\ 0$                   | $0\ 1$                | $1\ 0$                | $0\ 0$                |
| $1\ 0/1$                           | $0\ 0$                   | $0\ 1$                | $0\ 0$                | $0\ 0$                |
| $1\ 1/-$                           | $-$                      | $-$                   | $-$                   | $-$                   |

## - 2. mintafeladat -

### 5. A specifikációra érvényes vezérlési tábla elkészítése (Moore):

A D-MS saját  
vezérlési táblája

| $Q^n \rightarrow Q^{n+1}$ | D |
|---------------------------|---|
| $0 \rightarrow 0$         | 0 |
| $0 \rightarrow 1$         | 1 |
| $1 \rightarrow 0$         | 0 |
| $1 \rightarrow 1$         | 1 |

kódolt állapottábla

| kódolt aktuális<br>állapot/<br>kimenet | kódolt következő állapot |                       |                       |                       |
|----------------------------------------|--------------------------|-----------------------|-----------------------|-----------------------|
|                                        | $X_1 X_2$                |                       |                       |                       |
|                                        | $00$                     | $01$                  | $10$                  | $11$                  |
| $Q_1^n Q_2^n / Z$                      | $Q_1^{n+1} Q_2^{n+1}$    | $Q_1^{n+1} Q_2^{n+1}$ | $Q_1^{n+1} Q_2^{n+1}$ | $Q_1^{n+1} Q_2^{n+1}$ |
| $0\ 0/0$                               | $0\ 0$                   | $0\ 1$                | $0\ 0$                | $0\ 0$                |
| $0\ 1/0$                               | $0\ 0$                   | $0\ 1$                | $1\ 0$                | $0\ 0$                |
| $1\ 0/1$                               | $0\ 0$                   | $0\ 1$                | $0\ 0$                | $0\ 0$                |
| $1\ 1/-$                               | --                       | --                    | --                    | --                    |



vezérlési tábla

| kódolt aktuális<br>állapot/kimenet<br>$Q_1^n Q_2^n / Z$ | $X_1 X_2$ |           |           |           |
|---------------------------------------------------------|-----------|-----------|-----------|-----------|
|                                                         | $00$      | $01$      | $10$      | $11$      |
|                                                         | $D_1 D_2$ | $D_1 D_2$ | $D_1 D_2$ | $D_1 D_2$ |
| $0\ 0/0$                                                | $0\ 0$    | $0\ 1$    | $0\ 0$    | $0\ 0$    |
| $0\ 1/0$                                                | $0\ 0$    | $0\ 1$    | $1\ 0$    | $0\ 0$    |
| $1\ 0/1$                                                | $0\ 0$    | $0\ 1$    | $0\ 0$    | $0\ 0$    |
| $1\ 1/-$                                                | --        | --        | --        | --        |



## - 2. mintafeladat -

6. A szekunder változók és a kimenet függvényeinek Karnaugh tábla segítségével történő megadása (Moore):

**D1**

|       |       |         |   |   |   |   |
|-------|-------|---------|---|---|---|---|
|       |       | $Q_1^n$ | 0 | 0 | 1 | 1 |
|       |       | $Q_2^n$ | 0 | 1 | 1 | 0 |
| $X_1$ | $X_2$ |         |   |   |   |   |
| 0     | 0     |         |   | — |   |   |
| 0     | 1     |         |   | — |   |   |
| 1     | 1     |         |   | — |   |   |
| 1     | 0     |         |   | 1 | — |   |

$$D1 = Q_2^n X_1 \overline{X_2}$$

**D2**

|       |       |         |   |   |   |   |
|-------|-------|---------|---|---|---|---|
|       |       | $Q_1^n$ | 0 | 0 | 1 | 1 |
|       |       | $Q_2^n$ | 0 | 1 | 1 | 0 |
| $X_1$ | $X_2$ |         |   |   |   |   |
| 0     | 0     |         |   | — |   |   |
| 0     | 1     |         | 1 | 1 | — | 1 |
| 1     | 1     |         |   | — |   |   |
| 1     | 0     |         |   | — |   |   |

$$D2 = \overline{X_1} X_2$$

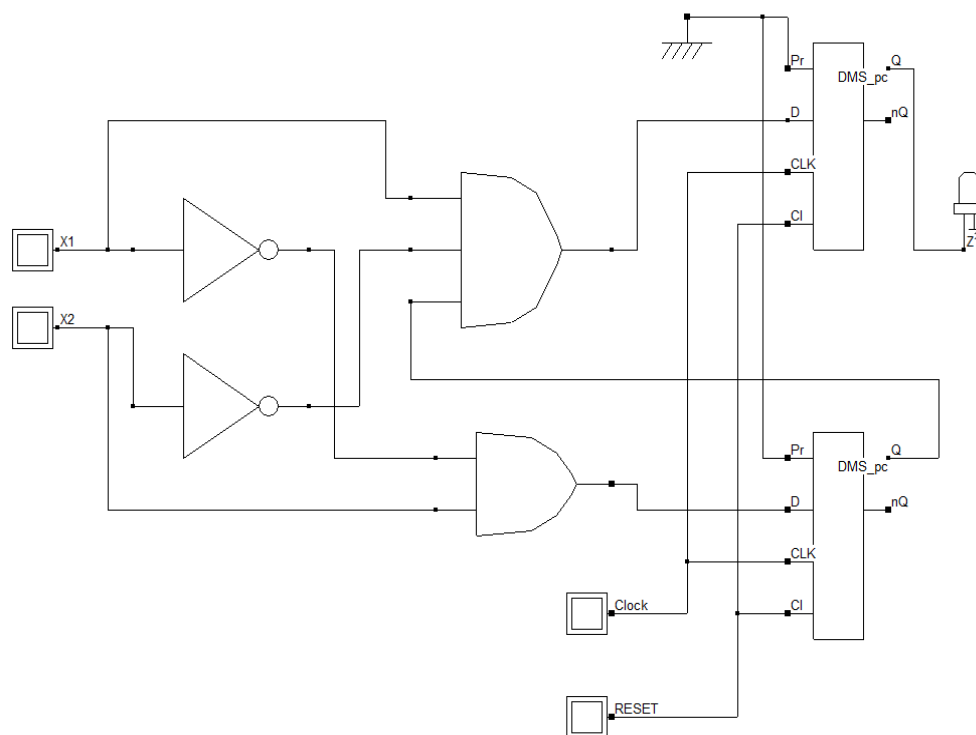
**Z**

|   |       |       |   |   |
|---|-------|-------|---|---|
|   |       | $Q_1$ | 0 | 1 |
|   | $Q_2$ |       |   |   |
| 0 |       |       |   | 1 |
| 1 |       |       |   | — |

$$Z = Q_1$$

## - 2. mintafeladat -

### 7-8. Realizáció, a kezdeti állapot biztosítása (RESET) (Moore):



$$D1 = Q_2^n X1 \overline{X2}$$

$$D2 = \overline{X1} X2$$

$$Z = Q_1$$

*A DSCH 3.5 szimulátorban megvalósított áramkör*

## - 2. mintafeladat -

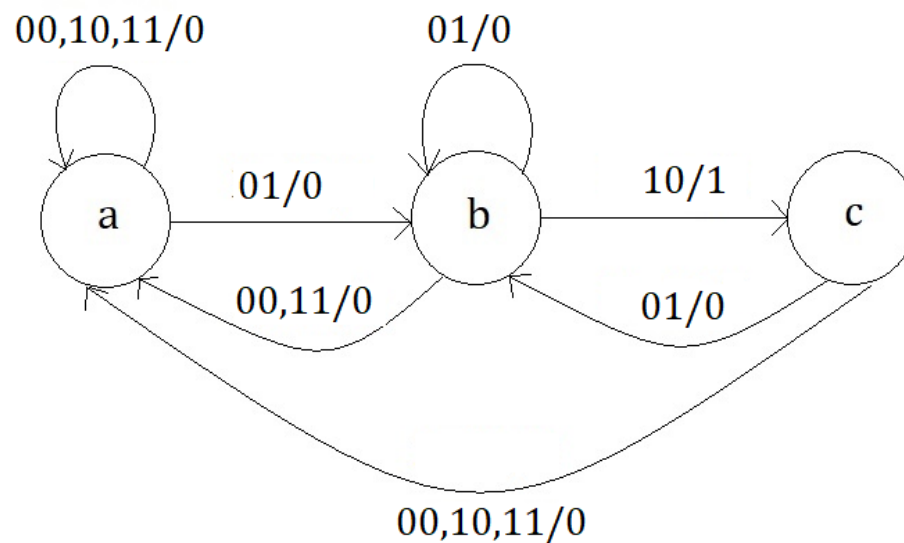
A feladat megoldásának(Moore) DSCH sémája  
(„SZH eloadasmintafeladat 2 dualishoz”):

<http://www.sze.hu/~somi/Digit%e1lis%20h%e1l%ef3zatok/DSCH%20s%e9m%e1k%20gyakorl%ef3%20feladatokhoz/>

## - 2. mintafeladat -

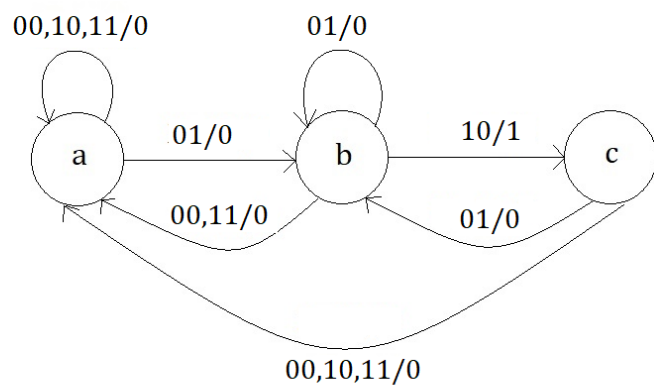
### Megoldás 2.(Mealy-modell):

#### 1. Az állapotgráf felrajzolása (Mealy):



## - 2. mintafeladat -

### 2. Az előzetes, szimbolikus állapottábla felvétele (Mealy):



*előzetes szimbolikus állapottábla*

| aktuális<br>állapot | következő állapot/kimenet |            |            |            |
|---------------------|---------------------------|------------|------------|------------|
|                     | $X_1X_2$                  |            |            |            |
|                     | <i>00</i>                 | <i>01</i>  | <i>10</i>  | <i>11</i>  |
| <i>a</i>            | <i>a/0</i>                | <i>b/0</i> | <i>a/0</i> | <i>a/0</i> |
| <i>b</i>            | <i>a/0</i>                | <i>b/0</i> | <i>c/1</i> | <i>a/0</i> |
| <i>c</i>            | <i>a/0</i>                | <i>b/0</i> | <i>a/0</i> | <i>a/0</i> |

## - 2. mintafeladat -

### 3. Összevont szimbolikus állapottábla megszerkesztése (Mealy):

*előzetes szimbolikus állapottábla*

| aktuális<br>Állapot | következő állapot/kimenet |            |            |            |
|---------------------|---------------------------|------------|------------|------------|
|                     | $X_1X_2$                  |            |            |            |
|                     | <i>00</i>                 | <i>01</i>  | <i>10</i>  | <i>11</i>  |
| <i>a</i>            | <i>a/0</i>                | <i>b/0</i> | <i>a/0</i> | <i>a/0</i> |
| <i>b</i>            | <i>a/0</i>                | <i>b/0</i> | <i>c/1</i> | <i>a/0</i> |
| <i>c</i>            | <i>a/0</i>                | <i>b/0</i> | <i>a/0</i> | <i>a/0</i> |

$$a, c \rightarrow ac$$

$$b \rightarrow b$$



*összevont szimbolikus állapottábla*

| aktuális<br>állapot | következő állapot/kimenet |            |             |             |
|---------------------|---------------------------|------------|-------------|-------------|
|                     | $X_1X_2$                  |            |             |             |
|                     | <i>00</i>                 | <i>01</i>  | <i>10</i>   | <i>11</i>   |
| <i>ac</i>           | <i>ac/0</i>               | <i>b/0</i> | <i>ac/0</i> | <i>ac/0</i> |
| <i>b</i>            | <i>ac/0</i>               | <i>b/0</i> | <i>ac/1</i> | <i>ac/0</i> |

## - 2. mintafeladat -

### 4. Kódolt állapottábla elkészítése (Mealy):

- Szekunder változó bevezetése az állapotok megkülönböztetésére:

egy lehetséges kódkiosztás:

$$ac \rightarrow Q=0$$

$$b \rightarrow Q=1$$



*kódolt állapottábla*

| aktuális<br>állapot,<br>$Q$ | következő állapot/kimenet |       |       |       |
|-----------------------------|---------------------------|-------|-------|-------|
|                             | $X_1X_2$                  |       |       |       |
|                             | $00$                      | $01$  | $10$  | $11$  |
| $0$                         | $0/0$                     | $1/0$ | $0/0$ | $0/0$ |
| $1$                         | $0/0$                     | $1/0$ | $0/1$ | $0/0$ |

## - 2. mintafeladat -

### 5. A specifikációra érvényes vezérlési tábla elkészítése (Mealy):

A D-MS saját  
vezérlési táblája

| $Q^n \rightarrow Q^{n+1}$ | D |
|---------------------------|---|
| $0 \rightarrow 0$         | 0 |
| $0 \rightarrow 1$         | 1 |
| $1 \rightarrow 0$         | 0 |
| $1 \rightarrow 1$         | 1 |

kódolt állapottábla

| aktuális<br>állapot,<br>$Q$ | $D/Z$    |     |     |     |
|-----------------------------|----------|-----|-----|-----|
|                             | $X_1X_2$ |     |     |     |
|                             | 00       | 01  | 10  | 11  |
| 0                           | 0/0      | 1/0 | 0/0 | 0/0 |
| 1                           | 0/0      | 1/0 | 0/1 | 0/0 |



vezérlési tábla

| aktuális<br>állapot,<br>$Q$ | $D/Z$    |     |     |     |
|-----------------------------|----------|-----|-----|-----|
|                             | $X_1X_2$ |     |     |     |
|                             | 00       | 01  | 10  | 11  |
| 0                           | 0/0      | 1/0 | 0/0 | 0/0 |
| 1                           | 0/0      | 1/0 | 0/1 | 0/0 |

## - 2. mintafeladat -

6. A szekunder változó és a kimenet függvényeinek Karnaugh tábla segítségével történő megadása (Mealy):

*kódolt állapottábla*

| aktuális<br>állapot,<br>$Q$ | $D/Z$    |       |       |       |
|-----------------------------|----------|-------|-------|-------|
|                             | $X_1X_2$ |       |       |       |
|                             | $00$     | $01$  | $10$  | $11$  |
| $0$                         | $0/0$    | $1/0$ | $0/0$ | $0/0$ |
| $1$                         | $0/0$    | $1/0$ | $0/1$ | $0/0$ |

| $D$ | $Q$ | $X_1$ | $0$ | $0$ | $1$ | $1$ |
|-----|-----|-------|-----|-----|-----|-----|
|     |     | $X_2$ | $0$ | $1$ | $1$ | $0$ |
|     | $0$ | $0$   | $1$ | $0$ | $0$ |     |
|     | $1$ | $0$   | $1$ | $0$ | $0$ |     |

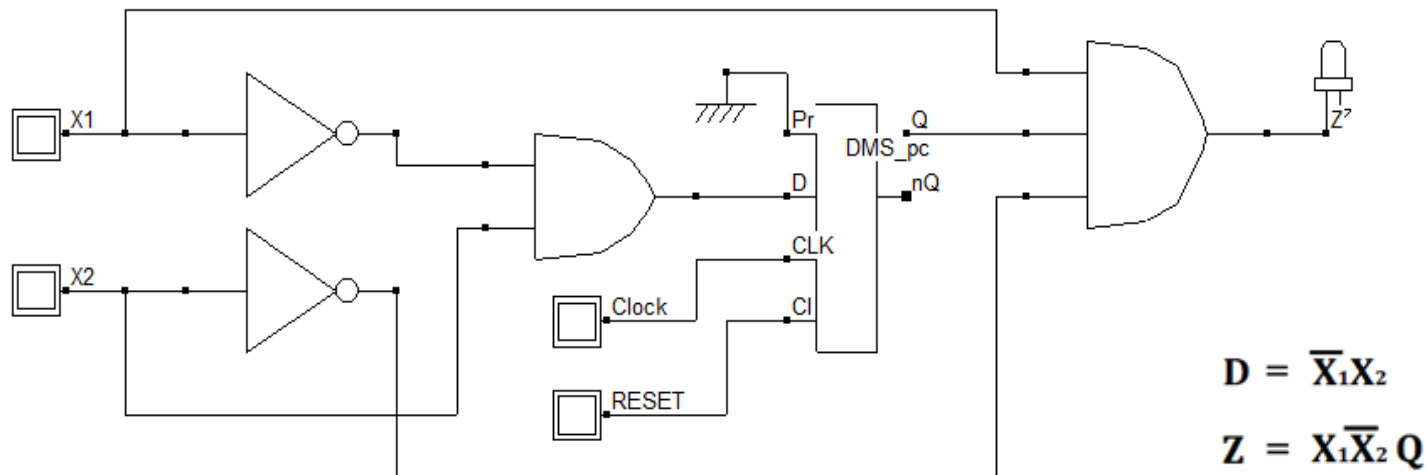
$$D = \bar{X}_1 X_2$$

| $Z$ | $Q$ | $X_1$ | $0$ | $0$ | $1$ | $1$ |
|-----|-----|-------|-----|-----|-----|-----|
|     |     | $X_2$ | $0$ | $1$ | $1$ | $0$ |
|     | $0$ | $0$   | $0$ | $0$ | $0$ |     |
|     | $1$ | $0$   | $0$ | $0$ | $1$ |     |

$$Z = X_1 \bar{X}_2 Q$$

## - 2. mintafeladat -

### 7-8. Realizáció, a kezdeti állapot biztosítása (RESET) (Mealy):



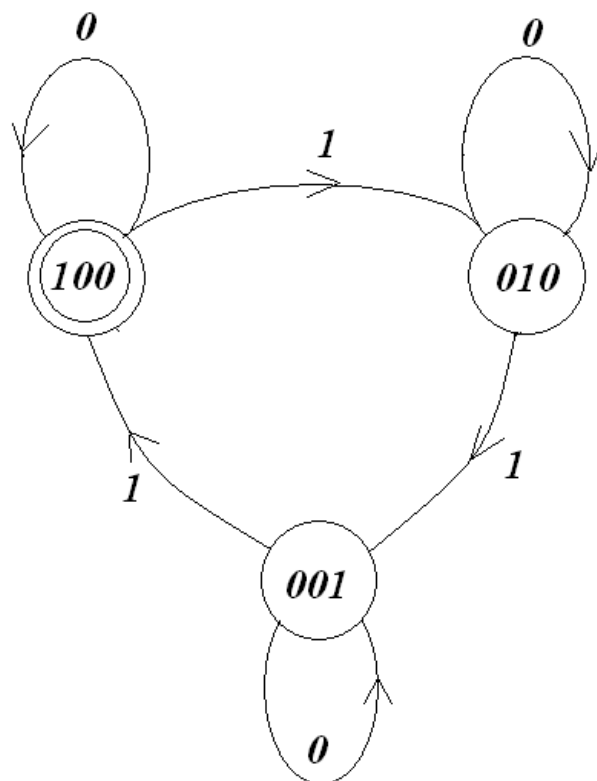
*A DSCH 3.5 szimulátorban megvalósított áramkör*

## Szinkron sorrendi hálózatok tervezése mintapéldákon keresztül I.

### - 3. mintafeladat -

*Tervezzük meg szinkron 'preset' és 'clear' bemenetekkel is rendelkező, élvezérelt **D-MS** tárolókkal az ábrán látható állapotgráf szerinti állapot-kimenetű szinkron sorrendi hálózatot a lehető legegyszerűbb változatban! A kezdeti állapotot a gráfon dupla kör jelzi.*

## - 3. mintafeladat -



## - 3. mintafeladat -

### **Megoldás 1.:**

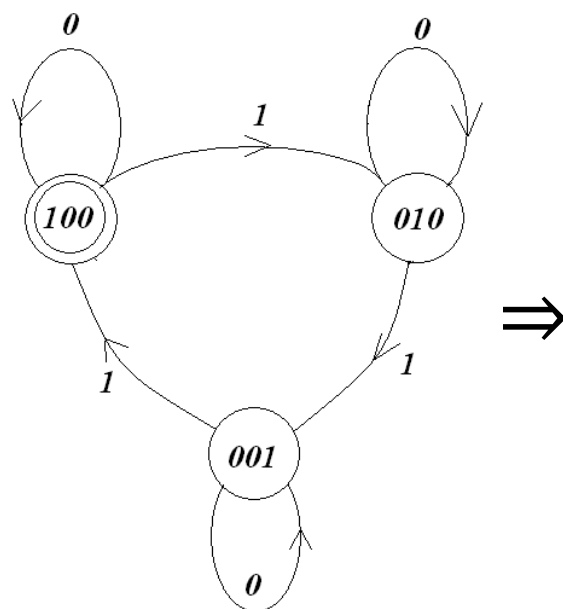
*A hálózat működési specifikációja a kódolt állapotgráffal lett megadva, így a megoldás első lépéseként rögtön elvégezhetjük a kódolt állapottábla felvételét.*

*Ugyanakkor az is látható a kódolt állapotgráfból, hogy az adott hálózat Moore-modell szerint működik.*

*A verbális specifikáció szerint a hálózat állapot-kimenetű: ez azt jelenti, hogy maguk az állapotokhoz tartozó kódok reprezentálják az aktuális kimeneti értékeket.*

## - 3. mintafeladat -

### 1. A kódolt állapottábla felvétele:



*kódolt állapottábla*

| kódolt aktuális<br>állapot, | kódolt következő állapot |         |         |             |             |             |
|-----------------------------|--------------------------|---------|---------|-------------|-------------|-------------|
|                             | X = 0                    |         |         | X = 1       |             |             |
|                             | $Q_1^n$                  | $Q_2^n$ | $Q_3^n$ | $Q_1^{n+1}$ | $Q_2^{n+1}$ | $Q_3^{n+1}$ |
| 1 0 0                       | 1                        | 0       | 0       | 1           | 0           | 0           |
| 0 1 0                       | 0                        | 1       | 0       | 0           | 1           | 0           |
| 0 0 1                       | 0                        | 0       | 1       | 0           | 0           | 1           |
| 0 0 0                       | 0                        | 0       | 0       | -           | -           | -           |
| 0 1 1                       | 0                        | 1       | 1       | -           | -           | -           |
| 1 0 1                       | 1                        | 0       | 1       | -           | -           | -           |
| 1 1 0                       | 1                        | 1       | 0       | -           | -           | -           |
| 1 1 1                       | 1                        | 1       | 1       | -           | -           | -           |

## - 3. mintafeladat -

### 2. A specifikációra érvényes vezérlési tábla elkészítése:

A D-MS saját  
vezérlési táblája

| $Q^n \rightarrow Q^{n+1}$ | D |
|---------------------------|---|
| $0 \rightarrow 0$         | 0 |
| $0 \rightarrow 1$         | 1 |
| $1 \rightarrow 0$         | 0 |
| $1 \rightarrow 1$         | 1 |



vezérlési tábla

| kódolt aktuális<br>állapot, | kódolt következő állapot |         |         |       |       |       |
|-----------------------------|--------------------------|---------|---------|-------|-------|-------|
|                             | X = 0                    |         |         | X = 1 |       |       |
|                             | $Q_1^n$                  | $Q_2^n$ | $Q_3^n$ | $D_1$ | $D_2$ | $D_3$ |
| 1 0 0                       | 1                        | 0       | 0       | 1     | 0     | 0     |
| 0 1 0                       | 0                        | 1       | 0       | 0     | 1     | 0     |
| 0 0 1                       | 0                        | 0       | 1       | 0     | 0     | 1     |
| 0 0 0                       | 0                        | 0       | 0       | -     | -     | -     |
| 0 1 1                       | 0                        | 1       | 1       | -     | -     | -     |
| 1 0 1                       | 1                        | 0       | 1       | -     | -     | -     |
| 1 1 0                       | 1                        | 1       | 0       | -     | -     | -     |
| 1 1 1                       | 1                        | 1       | 1       | -     | -     | -     |

## - 3. mintafeladat -

3. A szekunder változók ( $\rightarrow$ kimenetek) függvényeinek Karnaugh tábla segítségével történő megadása:

$\boxed{D_1}$

|           |     |     |     |     |
|-----------|-----|-----|-----|-----|
| $Q_1^n$   | 0   | 0   | 1   | 1   |
| $Q_2^n$   | 0   | 1   | 1   | 0   |
| $Q_3^n X$ | 0 0 | 0 1 | 1 1 | 1 0 |
|           | —   | —   | —   | 1   |
|           | —   | —   | —   | —   |
|           | 1   | —   | —   | —   |
|           | —   | —   | —   | —   |

$$D_1 = Q_1^n \bar{X} + Q_3^n X$$

$\boxed{D_2}$

|           |     |     |     |     |
|-----------|-----|-----|-----|-----|
| $Q_1^n$   | 0   | 0   | 1   | 1   |
| $Q_2^n$   | 0   | 1   | 1   | 0   |
| $Q_3^n X$ | 0 0 | 0 1 | 1 1 | 1 0 |
|           | —   | 1   | —   | —   |
|           | —   | —   | —   | 1   |
|           | —   | —   | —   | —   |
|           | —   | —   | —   | —   |

$$D_2 = Q_2^n \bar{X} + Q_1^n X$$

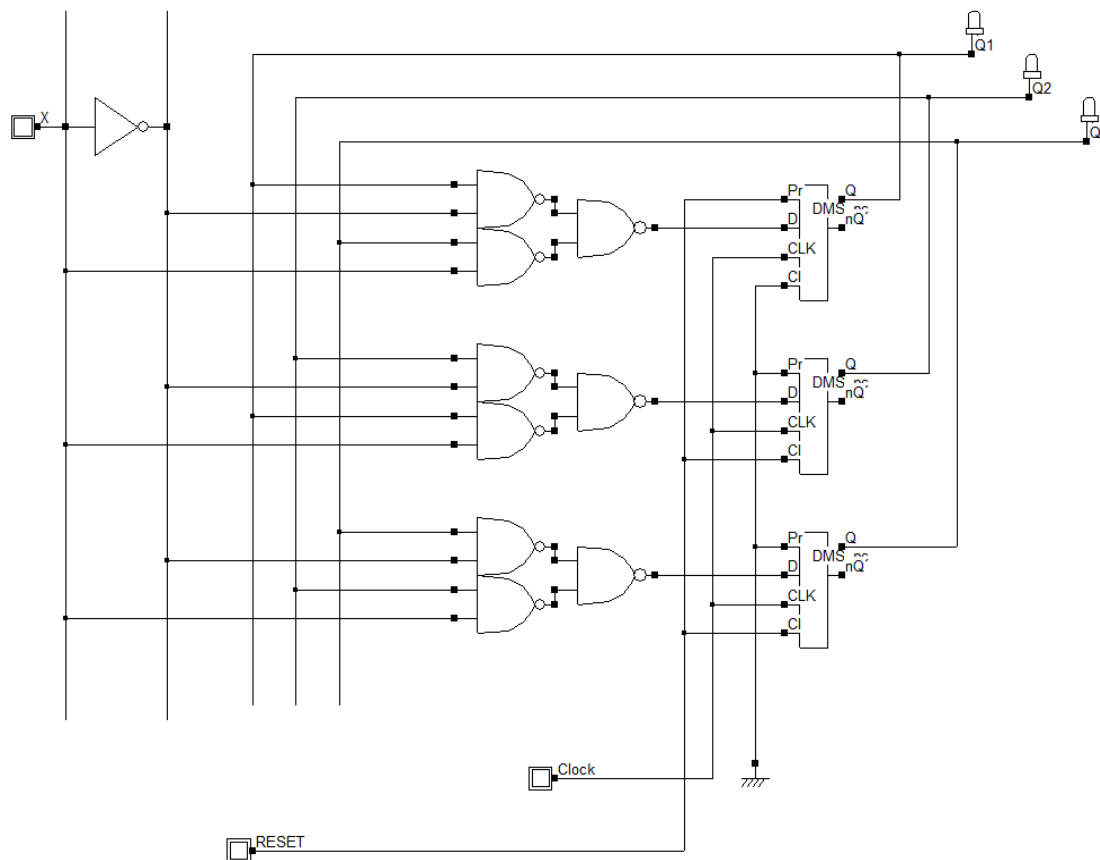
$\boxed{D_3}$

|           |     |     |     |     |
|-----------|-----|-----|-----|-----|
| $Q_1^n$   | 0   | 0   | 1   | 1   |
| $Q_2^n$   | 0   | 1   | 1   | 0   |
| $Q_3^n X$ | 0 0 | 0 1 | 1 1 | 1 0 |
|           | —   | —   | —   | —   |
|           | —   | 1   | —   | —   |
|           | —   | —   | —   | —   |
|           | 1   | —   | —   | —   |

$$D_3 = Q_2^n X + Q_3^n \bar{X}$$

## - 3. mintafeladat -

### 4. Realizáció(NAND'), a kezdeti állapot biztosítása (RESET):



$$D_1 = Q_1^n \bar{X} + Q_3^n X$$

$$D_2 = Q_2^n \bar{X} + Q_1^n X$$

$$D_3 = Q_2^n X + Q_3^n \bar{X}$$

A DSCH 3.5 szimulátorban megvalósított áramkör

## - 3. mintafeladat -

A feladat megoldásának DSCH sémája  
(„SZH eloadasmintafeladat 3 dualishoz”):

<http://www.sze.hu/~somi/Digit%e1lis%20h%e1l%3zatok/DSCH%20s%e9m%e1k%20gyakorl%3%20feladatokhoz/>

## - 3. mintafeladat -

### **Megoldás 2. (,1'-es súlyozású kódolási alap):**

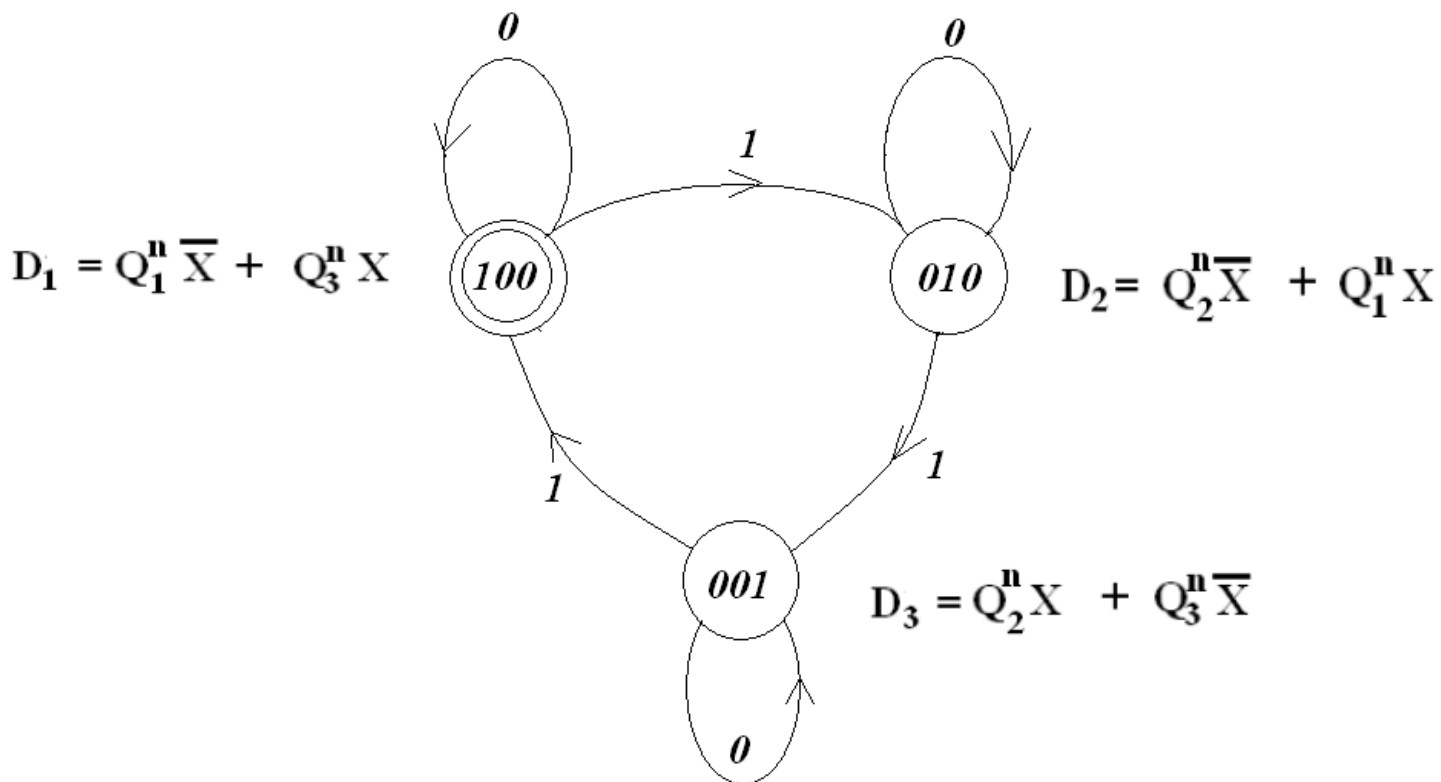
*A megoldástípus alapja az ,1'-es súlyozású kódolás:  
az olyan kódolást, melyben csak egyetlen bit helyén szerepel '1'-es  
érték (a többi '0'), ,1'-es súlyú kódolásnak nevezzük.*

Vegyük észre! ☺ :

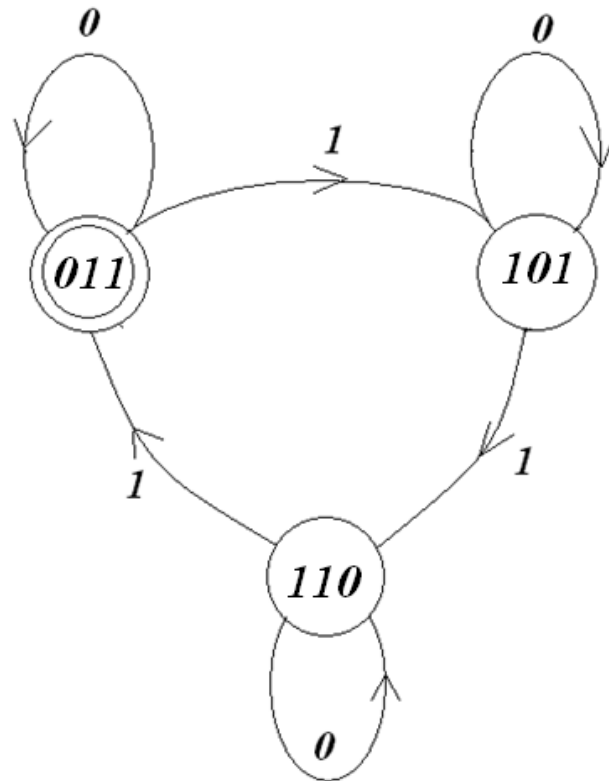
a **D** bemenetek vezérlésének megvalósítása ilyenkor rendkívül egyszerű, és az állapotgráf alapján elvégezhető: minden egyes **D** bemenetre akkor és csak akkor kell '1'-et kapcsolni, amikor az általa reprezentált tároló kimenetnek '0'-ról '1'-re kell változnia, azaz amikor az adott flip-flop által reprezentált állapotnak be kell állnia. Ez pedig az állapotgráfból egyértelműen kiolvasható. Az így kapott algebrai függvényalak megegyezik a Karnaugh táblák alapján meghatározottakkal.

- 3. mintafeladat -

**Megoldás 2. (,1'-es súlyozású kódolási alap):**



Példa egy hasonló, de nem ,1'-es súlyozású specifikációra:



?

Vegyük észre! ☺ ....

## Szinkron sorrendi hálózatok tervezése mintapéldákon keresztül I.

### - 4. mintafeladat -

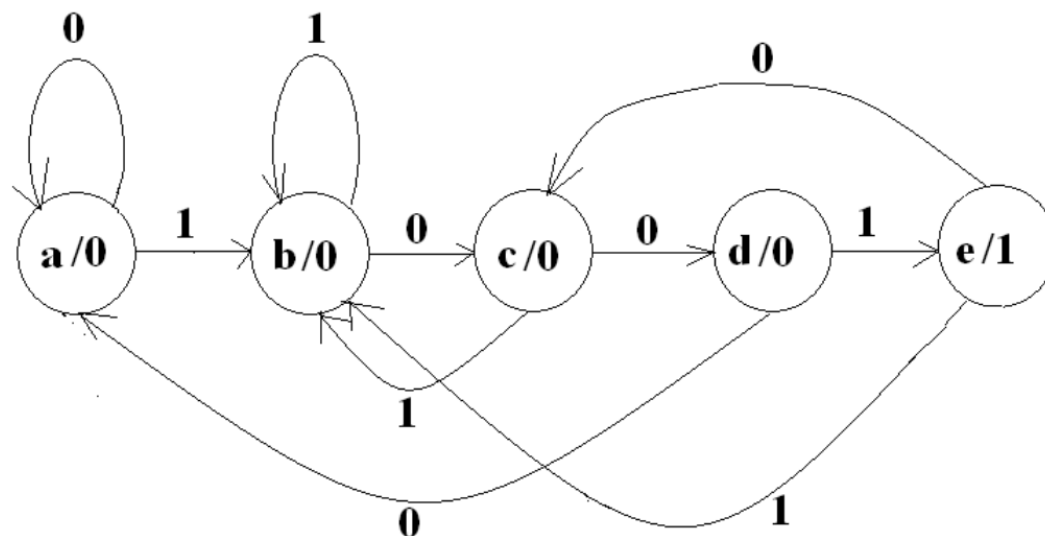
*Tervezzük meg egyes súlyú állapotkóddal, „Preset” és „Clear” bemenettel is rendelkező D-MS flip-flopokkal azt az egybemenetű (X) és egykimenetű (Z), Moore-típusú szinkron sorrendi hálózatot, amely  $Z=1$  jelzéssel mutatja meg az ...-1-0-0-1-... bemeneti sorozat megjelenését egy soros bemeneti szalagon!*

*(Hf.: tervezzük meg a feladatot intuitív módon („szabadon”) kódolt szekunder változókkal is!)*

## - 4. mintafeladat -

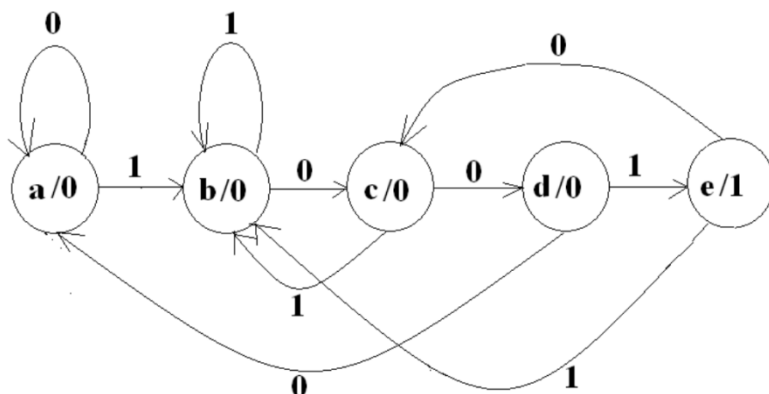
### Megoldás:

#### 1. Az állapotgráf felrajzolása:



## - 4. mintafeladat -

### 2. '1'-es súlyozású kódolás:



- egy fiktív kódolás:

|   | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | $Q_5$ |
|---|-------|-------|-------|-------|-------|
| a | 1     | 0     | 0     | 0     | 0     |
| b | 0     | 1     | 0     | 0     | 0     |
| c | 0     | 0     | 1     | 0     | 0     |
| d | 0     | 0     | 0     | 1     | 0     |
| e | 0     | 0     | 0     | 0     | 1     |



### 3. Függvények felírása:

$$D_1 = Q_1^n \overline{X} + Q_4^n \overline{X}$$

$$= (Q_1^n + Q_4^n) \overline{X}$$

$$D_2 = Q_1^n X + Q_2^n X + Q_3^n X + Q_5^n X$$

$$= \overline{Q_4^n} X$$

$$D_3 = Q_2^n \overline{X} + Q_5^n \overline{X}$$

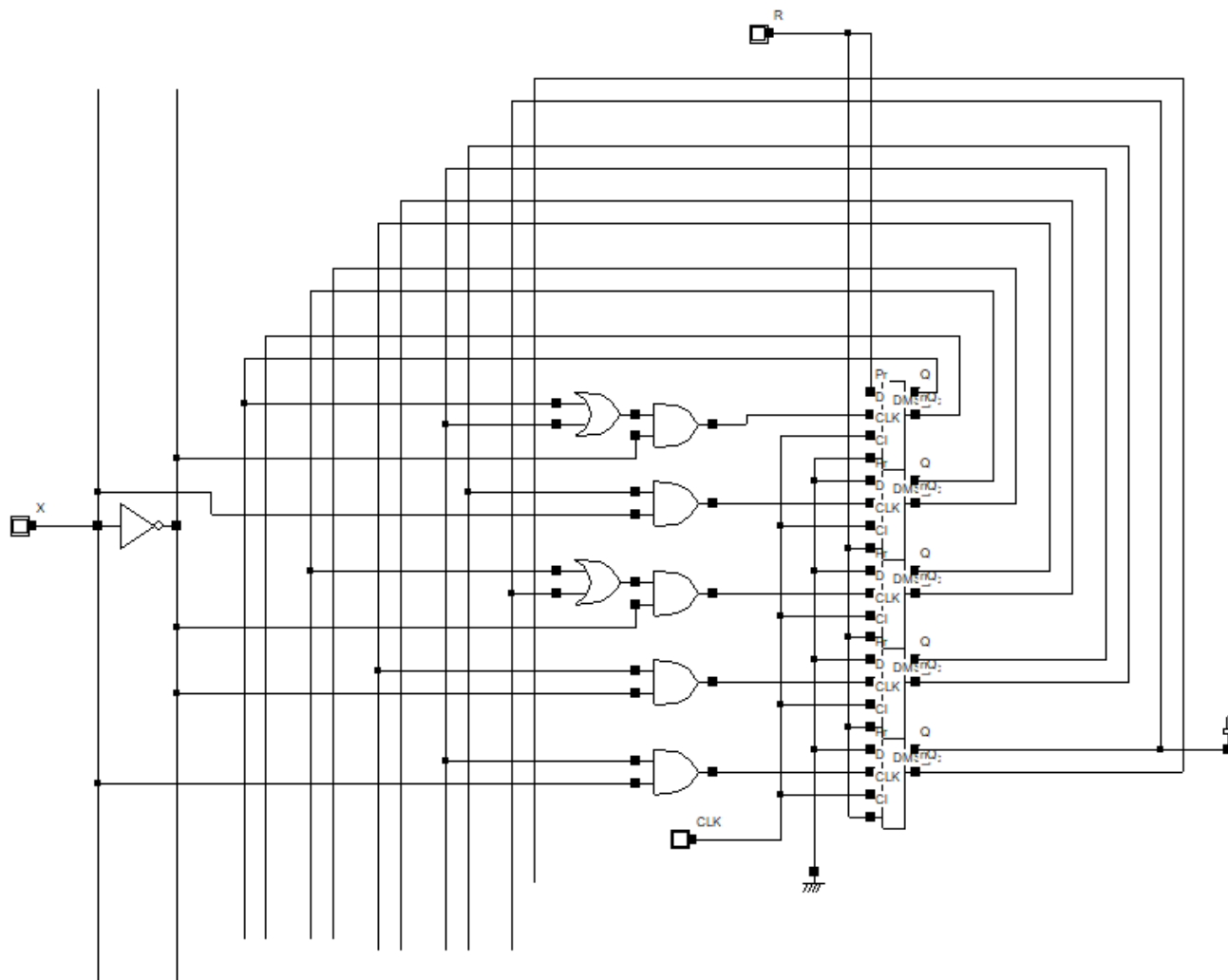
$$= (Q_2^n + Q_5^n) \overline{X}$$

$$D_4 = Q_3^n \overline{X}$$

$$D_5 = Q_4^n X$$

## - 4. mintafeladat -

### 4. Realizáció a kezdeti állapot biztosításával (RESET):



$$D_1 = Q_1^n \overline{X} + Q_4^n \overline{X}$$

$$= (Q_1^n + Q_4^n) \overline{X}$$

$$D_2 = Q_1^n X + Q_2^n X + Q_3^n X + Q_5^n X$$

$$= \overline{Q_4^n} X$$

$$D_3 = Q_2^n \overline{X} + Q_5^n \overline{X}$$

$$= (Q_2^n + Q_5^n) \overline{X}$$

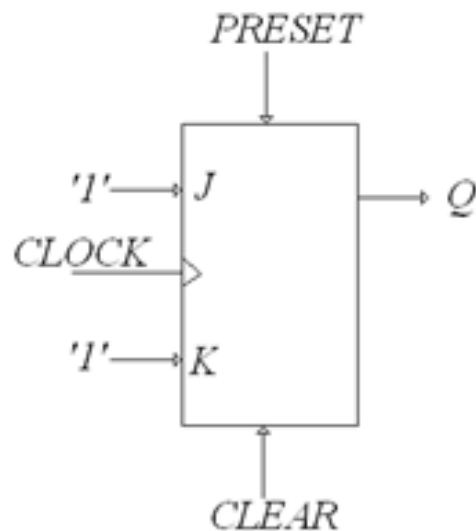
$$D_4 = Q_3^n \overline{X}$$

$$D_5 = Q_4^n X$$

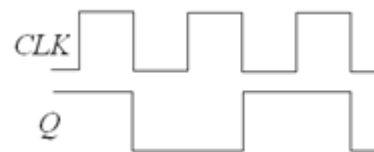
## Szinkron sorrendi hálózatok tervezése mintapéldákon keresztül II.:

### számlálókkal felépített szinkron sorrendi hálózatok tervezése

A **JK-MS** tároló, mint a számláló alapeleme: a kettes osztó funkció



$$JK(,11') \rightarrow Q^{n+1} = \overline{Q^n}$$

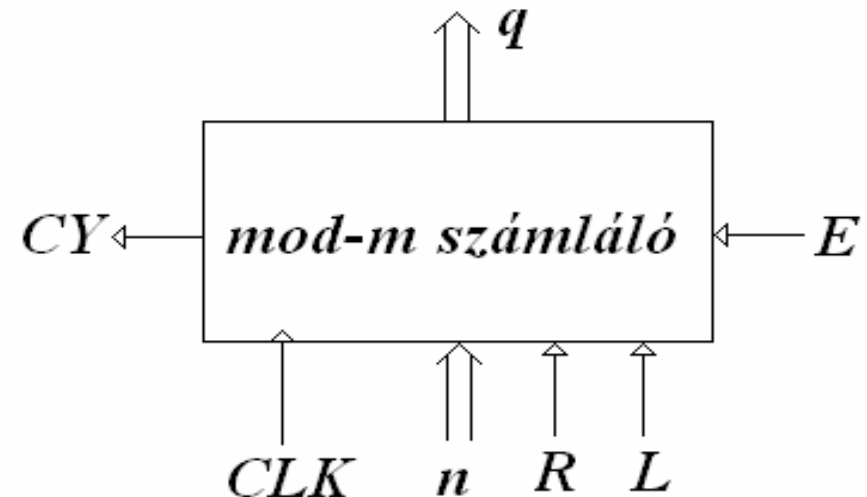


## Szinkron számlálók

szinkron *mod-m* számláló általános szimbóluma:

### bemenetek:

- *R*: reset
  - *L*: load
  - *E*: enable
  - *CLK*: clock
  - *n*: bitvektor adatbemenetek
- } vezérlőbemenetek,  
prioritás: *R*, *L*, *E*



### kimenetek:

- *q*: adatkimenetek(„szám”)
- *CY*: carry

## Szinkron számlálók

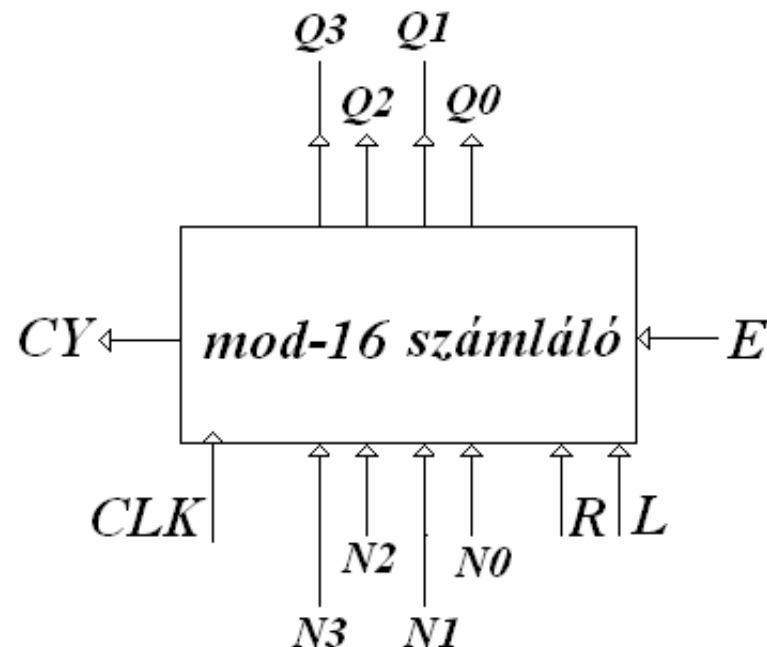
Példa: szinkron mod-16 számláló szimbóluma:

### bemenetek:

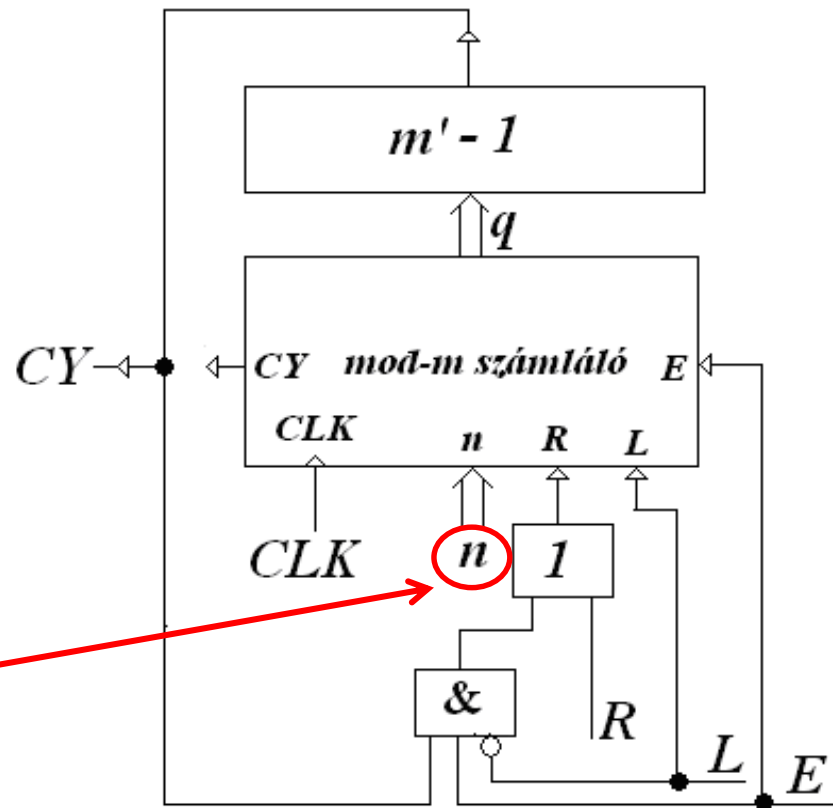
- $R$ : reset
  - $L$ : load
  - $E$ : enable
  - $CLK$ : clock
  - $N3-N0$ : bitvektor adatbemenetek
- } vezérlőbemenetek,  
prioritás:  $R, L, E$

### kimenetek:

- $Q3-Q0$ : adatkimenetek („szám”)
- $CY$ : carry

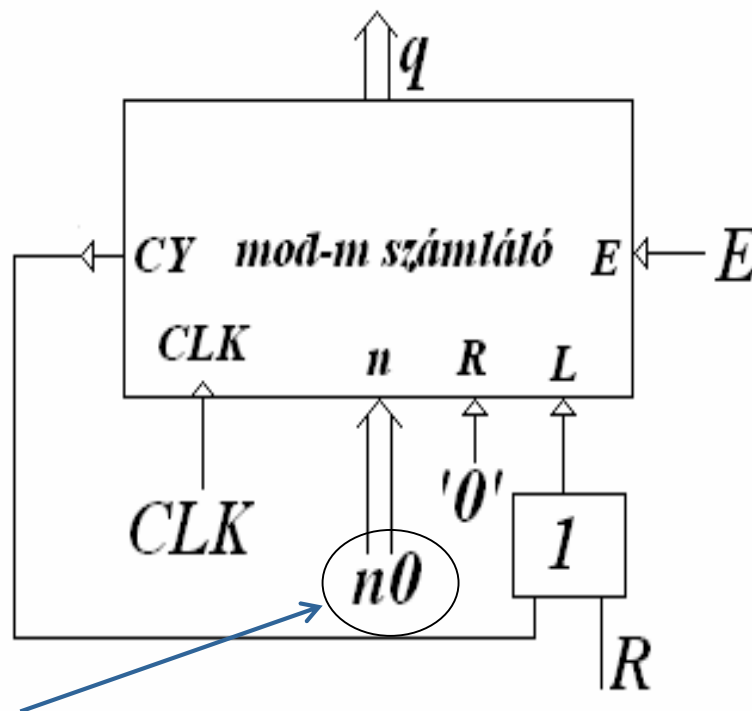


## Adott modulusú számláló átalakítása más modulusúvá

$$m' < m$$
$$n \leq m'-1 !!$$


## Szinkron számlálók

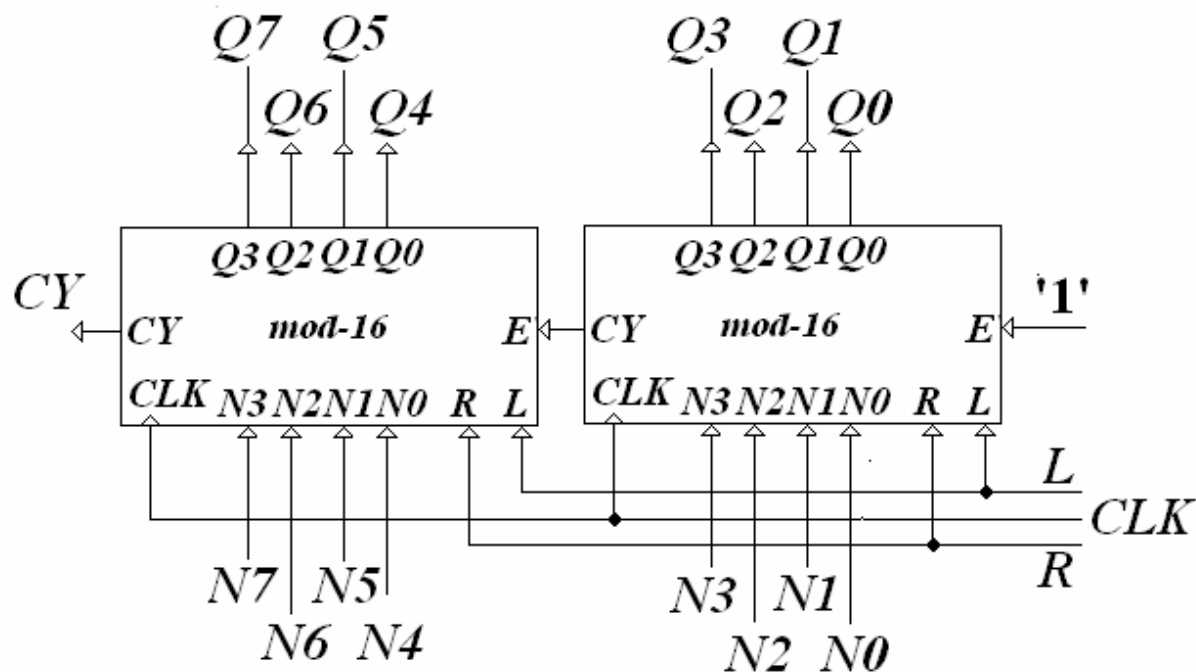
Számláló nullától különböző kezdőértékének beállítása



$n0$  : „kezdőérték”

## Szinkron számlálók

Példa: mod-256-os számláló mod-16 számlálókból

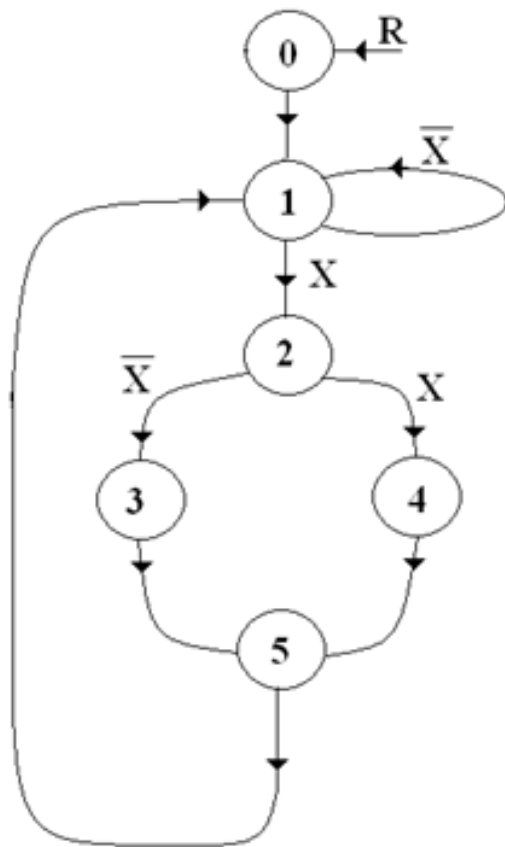


## Szinkron sorrendi hálózat tervezése szinkron számlálóval

- 1. mintafeladat -

*Valósítsuk meg törölhető (R), betölthető (L) és engedélyezhető (E) számlálóval és multiplexerekkel a mellékelt kódolt állapotgráf szerinti szinkron sorrendi hálózatot!*

## - 1. mintafeladat -

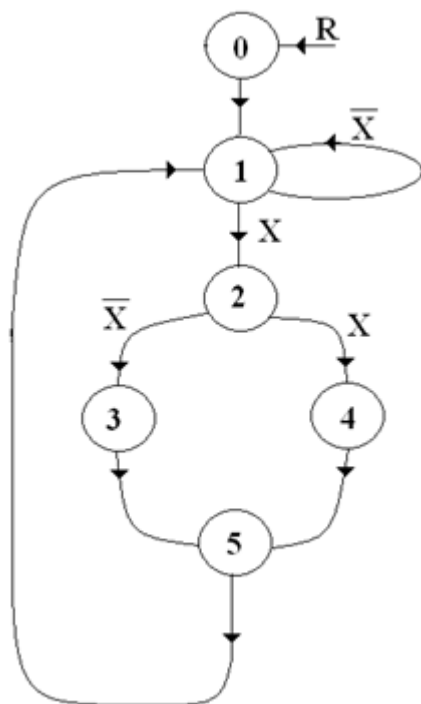


állapot kimenetű kódolt állapotgráf

## - 1. mintafeladat -

### Megoldás:

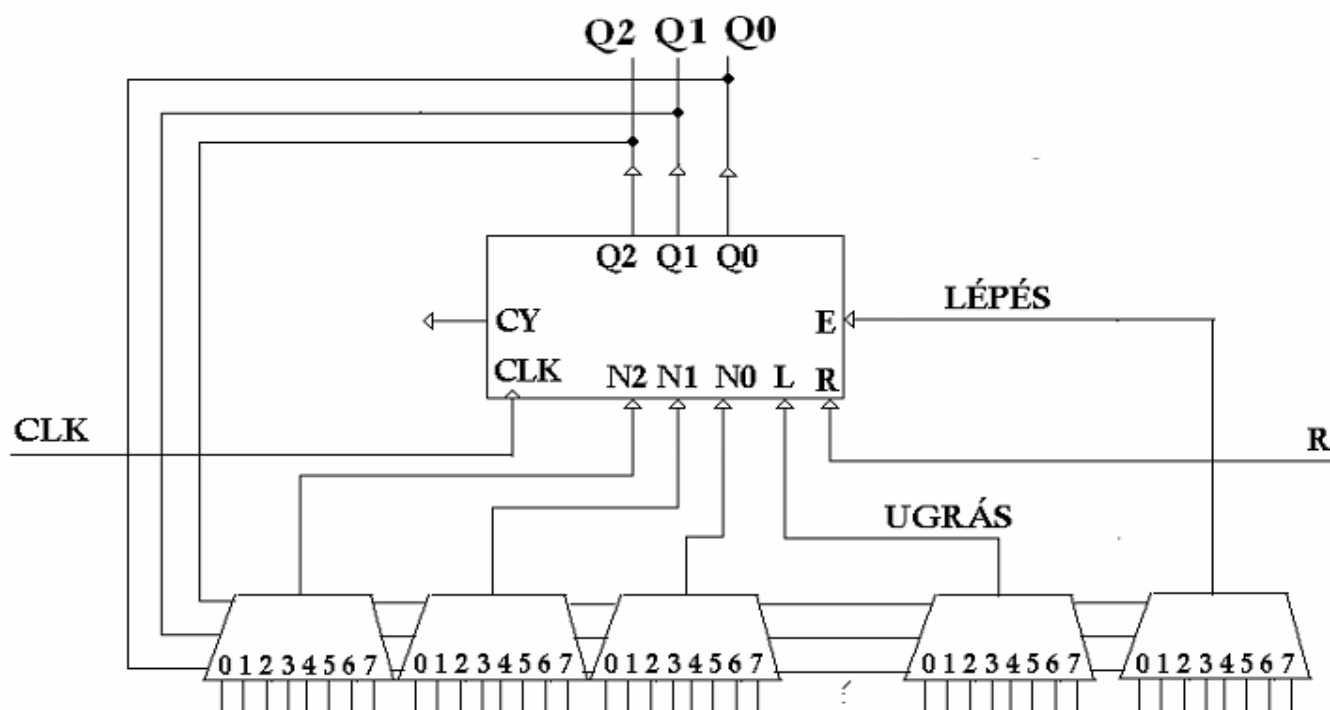
*Táblázatok a megvalósításhoz:*



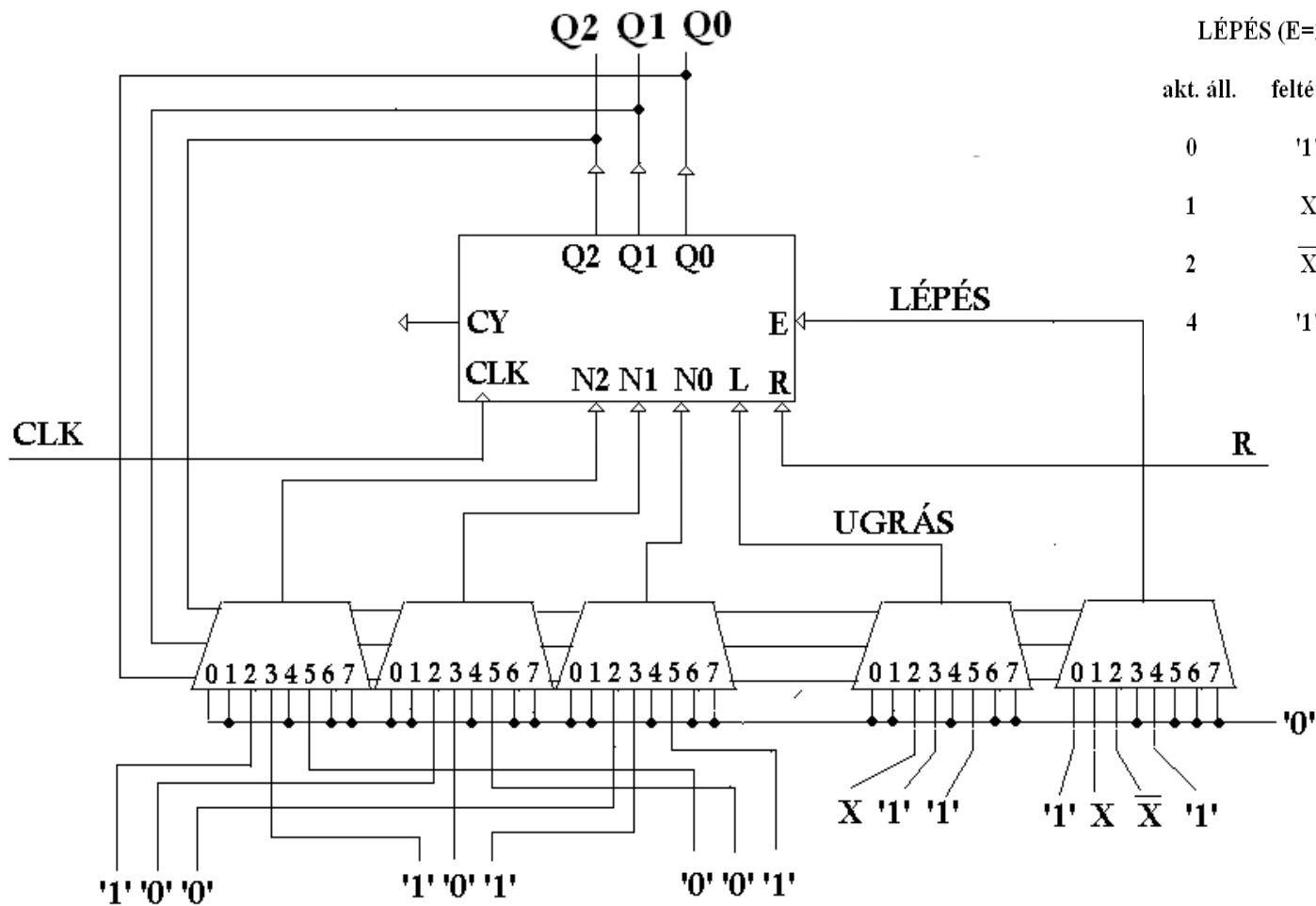
| LÉPÉS (E=1) |                | UGRÁS (L=1) |          |          |
|-------------|----------------|-------------|----------|----------|
| akt. áll.   | feltétel       | akt.áll.    | feltétel | köv.áll. |
| 0           | '1'            | 2           | X        | 4        |
| 1           | X              | 3           | '1'      | 5        |
| 2           | $\overline{X}$ | 5           | '1'      | 1        |
| 4           | '1'            |             |          |          |

## - 1. mintafeladat -

### Speciális célarchitektúra szinkron sorrendi hálózatok számlálós megvalósítására



- 1. mintafeladat -



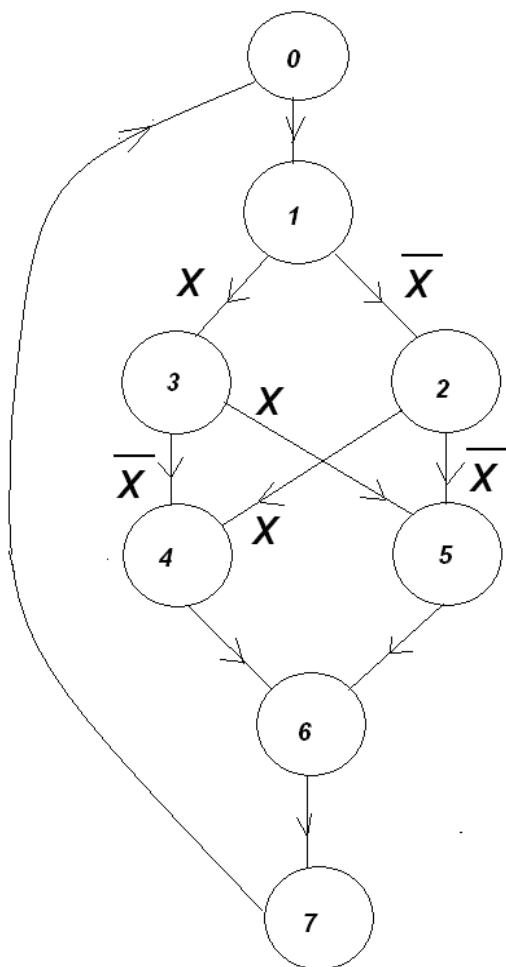
| LÉPÉS (E=1) |                | UGRÁS (L=1) |          |          |
|-------------|----------------|-------------|----------|----------|
| akt. áll.   | feltétel       | akt.áll.    | feltétel | köv.áll. |
| 0           | '1'            | 2           | X        | 4        |
| 1           | X              | 3           | '1'      | 5        |
| 2           | $\overline{X}$ | 5           | '1'      | 1        |
| 4           | '1'            |             |          |          |

## Szinkron sorrendi hálózat tervezése szinkron számlálóval

- 2. mintafeladat -

*Valósítsuk meg törölhető (R), betölthető (L) és engedélyezhető (E) számlálóval és multiplexerekkel a mellékelt kódolt állapotgráf szerinti szinkron sorrendi hálózatot!*

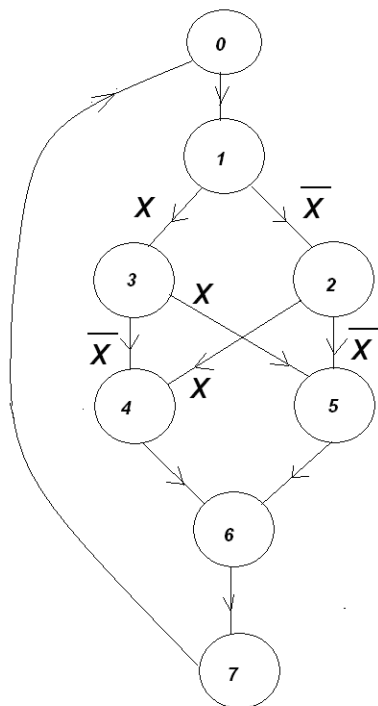
## - 2. mintafeladat -



állapot kimenetű kódolt állapotgráf

## - 2. mintafeladat -

### Megoldás:



### Táblázatok a megvalósításhoz:

| lépések  |                | ugrások      |                |              |
|----------|----------------|--------------|----------------|--------------|
| akt. áll | feltétel       | akt. állapot | feltétel       | köv. állapot |
| 0        | '1'            | 1            | X              | 3            |
| 1        | $\overline{X}$ | 2            | $\overline{X}$ | 5            |
| 3        | $\overline{X}$ | 2            | X              | 4            |
| 5        | '1'            | 3            | X              | 5            |
| 6        | '1'            | 4            | '1'            | 6            |
| 7        | '1'            |              |                |              |

## - 2. mintafeladat -

*Kettős ugrás az „ugrások” táblában:*

| lépések  |                | ugrások      |                |              |
|----------|----------------|--------------|----------------|--------------|
| akt. áll | feltétel       | akt. állapot | feltétel       | köv. állapot |
| 0        | '1'            | 1            | X              | 3            |
| 1        | $\overline{X}$ | 2            | $\overline{X}$ | 5            |
| 3        | $\overline{X}$ | 2            | X              | 4            |
| 5        | '1'            | 3            | X              | 5            |
| 6        | '1'            | 4            | '1'            | 6            |
| 7        | '1'            |              |                |              |

## - 2. mintafeladat -

Megoldás a kettős ugrás kezelésére:

| akt. állapot | feltétel       | köv. állapot |  |  |  |  |
|--------------|----------------|--------------|--|--|--|--|
| 2            | $\overline{X}$ | 5            |  |  |  |  |
| 2            | $X$            | 4            |  |  |  |  |

$\Rightarrow$

| segédtáblázat a kettős ugrás<br>lekezelésére |    |    |    |  |
|----------------------------------------------|----|----|----|--|
|                                              | N2 | N1 | N0 |  |
| $\overline{X}$                               | 1  | 0  | 1  |  |
| $X$                                          | 1  | 0  | 0  |  |

Ha  $q = 2$ ,

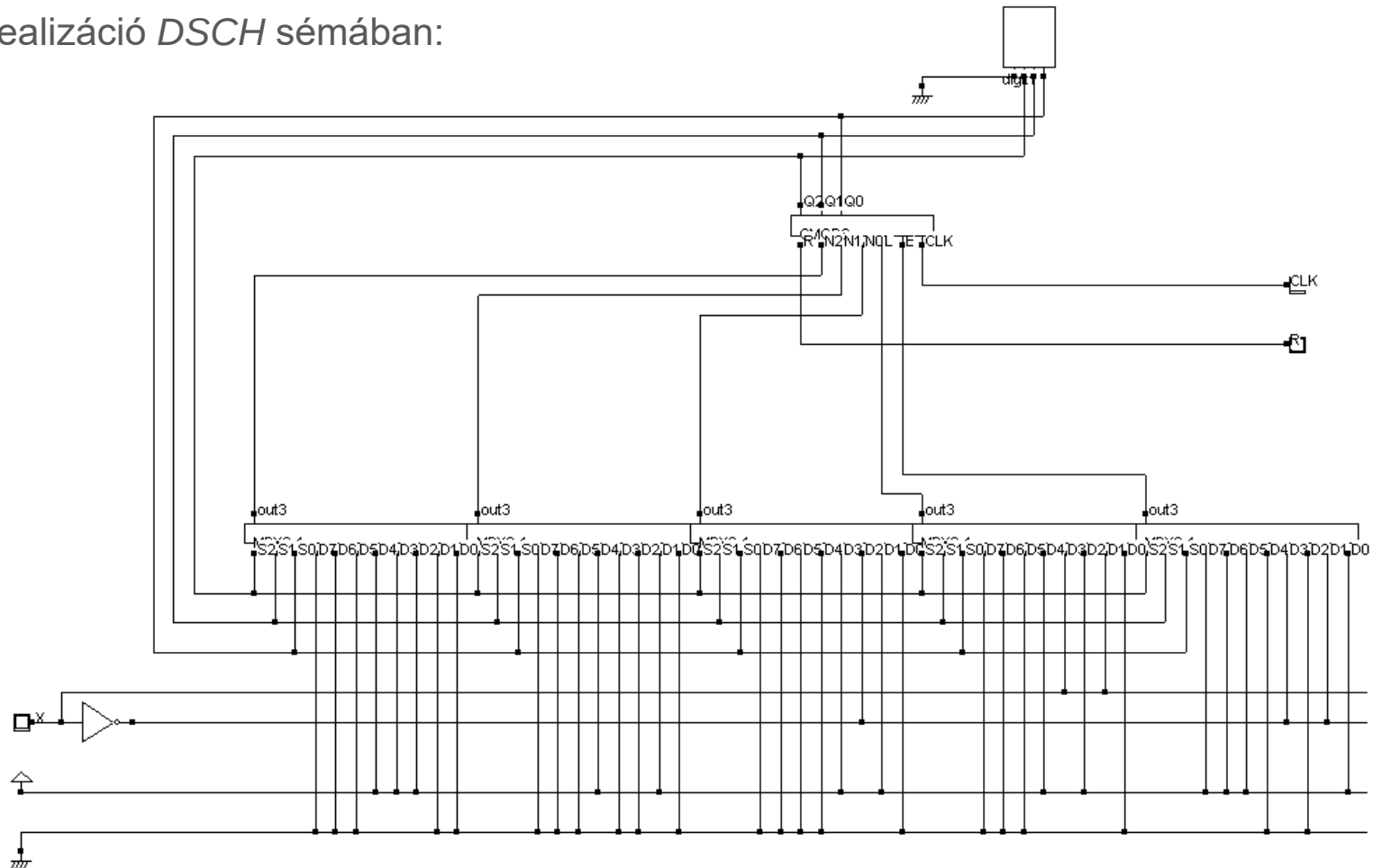
$$N2(D2) = 1$$

$$N1(D2) = 0$$

$$N0(D2) = \overline{X}$$

## - 2. mintafeladat -

Realizáció *DSCH* sémában:



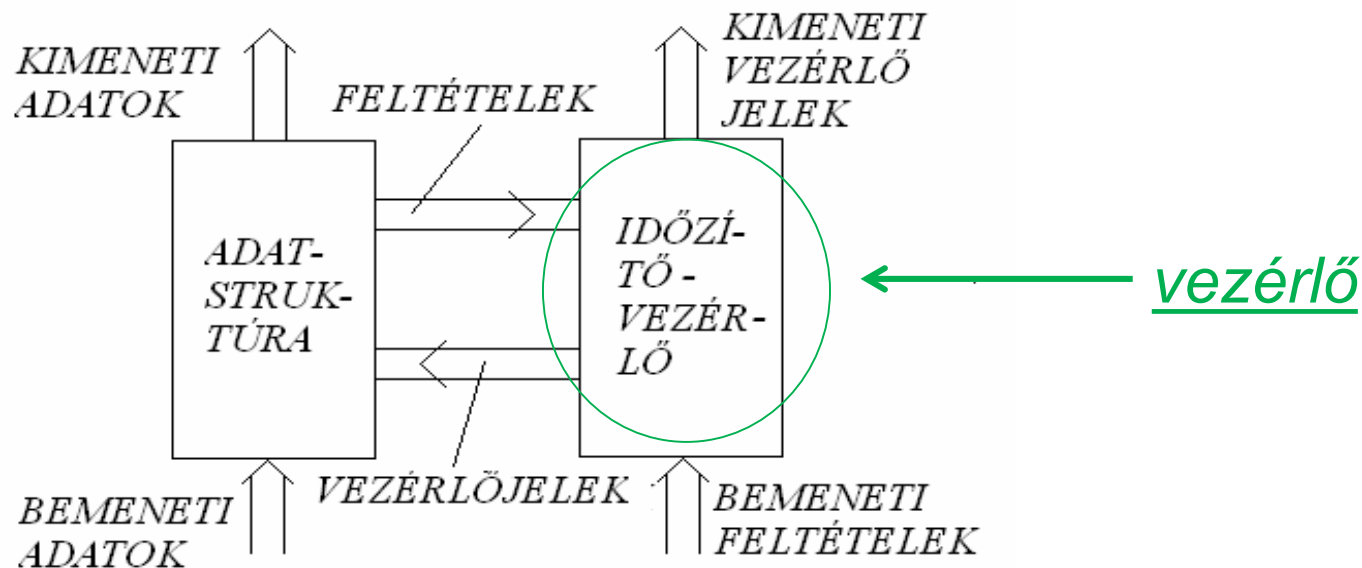
## - 2. mintafeladat -

A feladat megoldásának DSCH sémája  
(„SZH számláló előadásmintafeladat 2 dualishoz”):

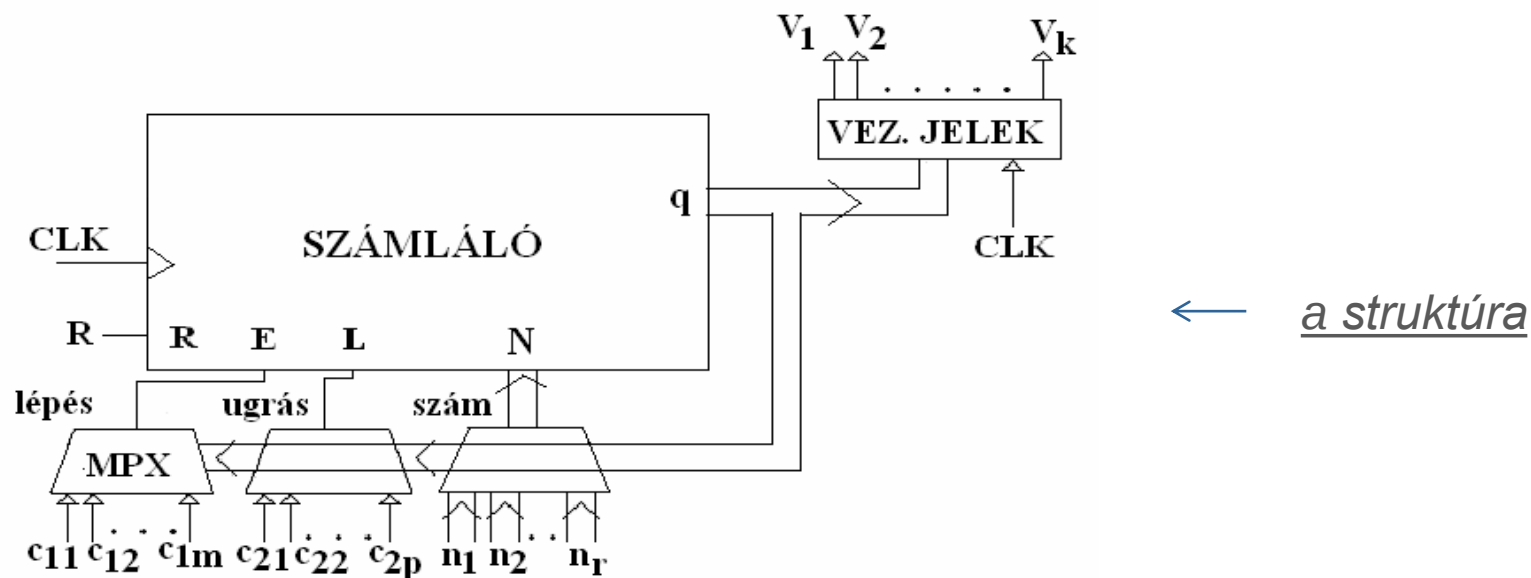
<http://www.sze.hu/~somi/Digit%e1lis%20h%e1l%3zatok/DSCH%20s%e9m%e1k%20gyakorl%3%20feladatokhoz/>

## Szinkron sorrendi hálózat tervezése szinkron számlálóval: szinkron számlálók alkalmazása időzítő-vezérlő egységekben

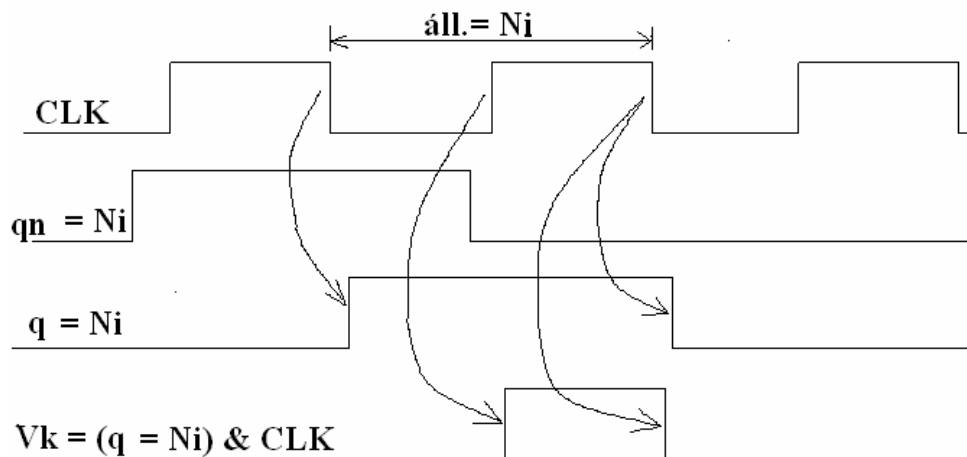
*A digitális egység felbontása adat- és  
vezérlő-alegységre :*



## Számláló-típusú vezérlők



A hazardmentes vezérlés feltétele:

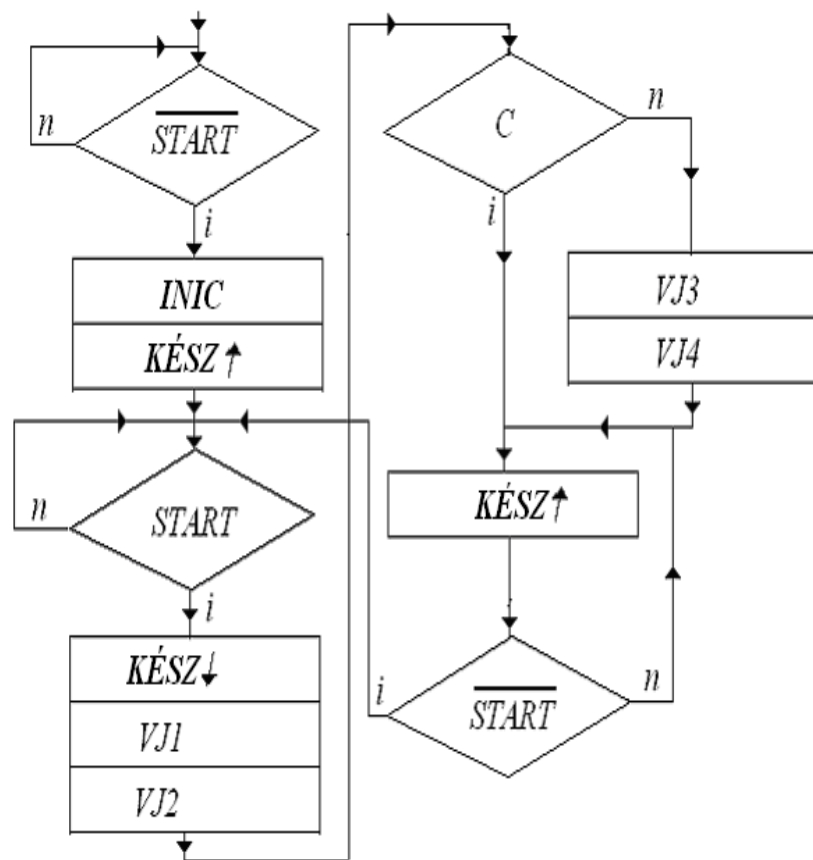


## Szinkron sorrendi hálózat tervezése szinkron számlálóval

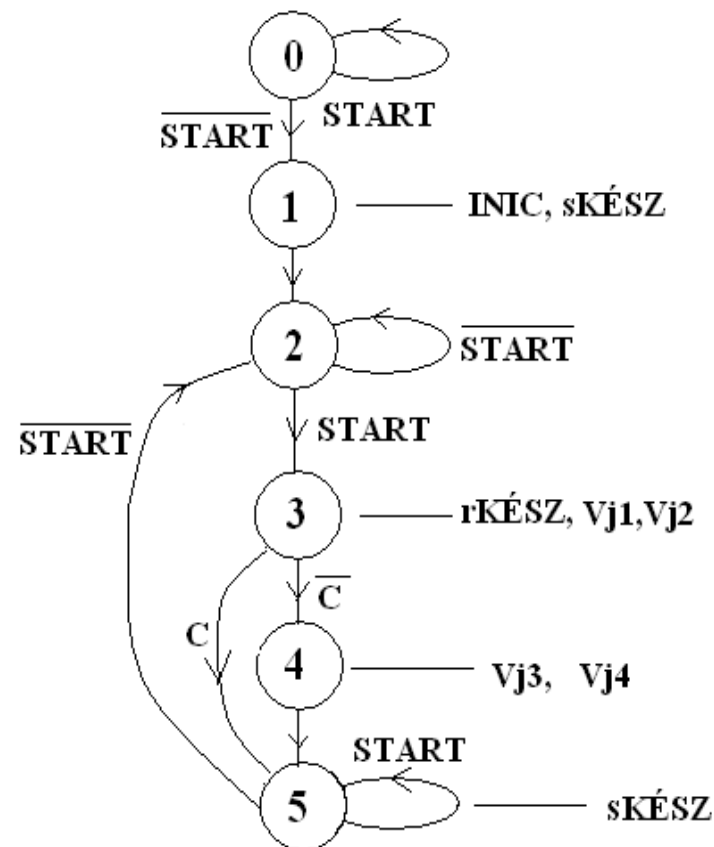
- 3. mintafeladat -

*Tervezzünk időzítő-vezérlő egységet szinkron számláló felhasználásával a mellékelt folyamatábra alapján!*

## - 3. mintafeladat -

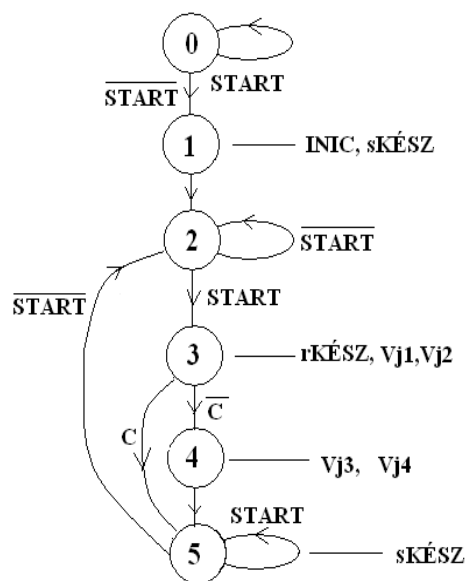


$\Rightarrow$

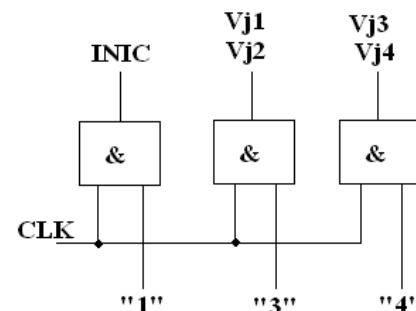
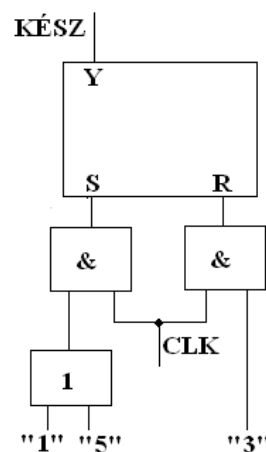
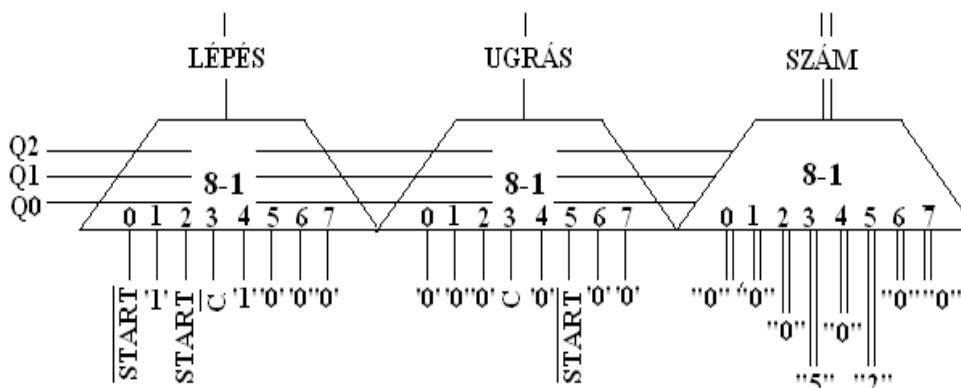


## - 3. mintafeladat -

### Megoldás:



állapot-kimenetű kódolt  
állapotgráf

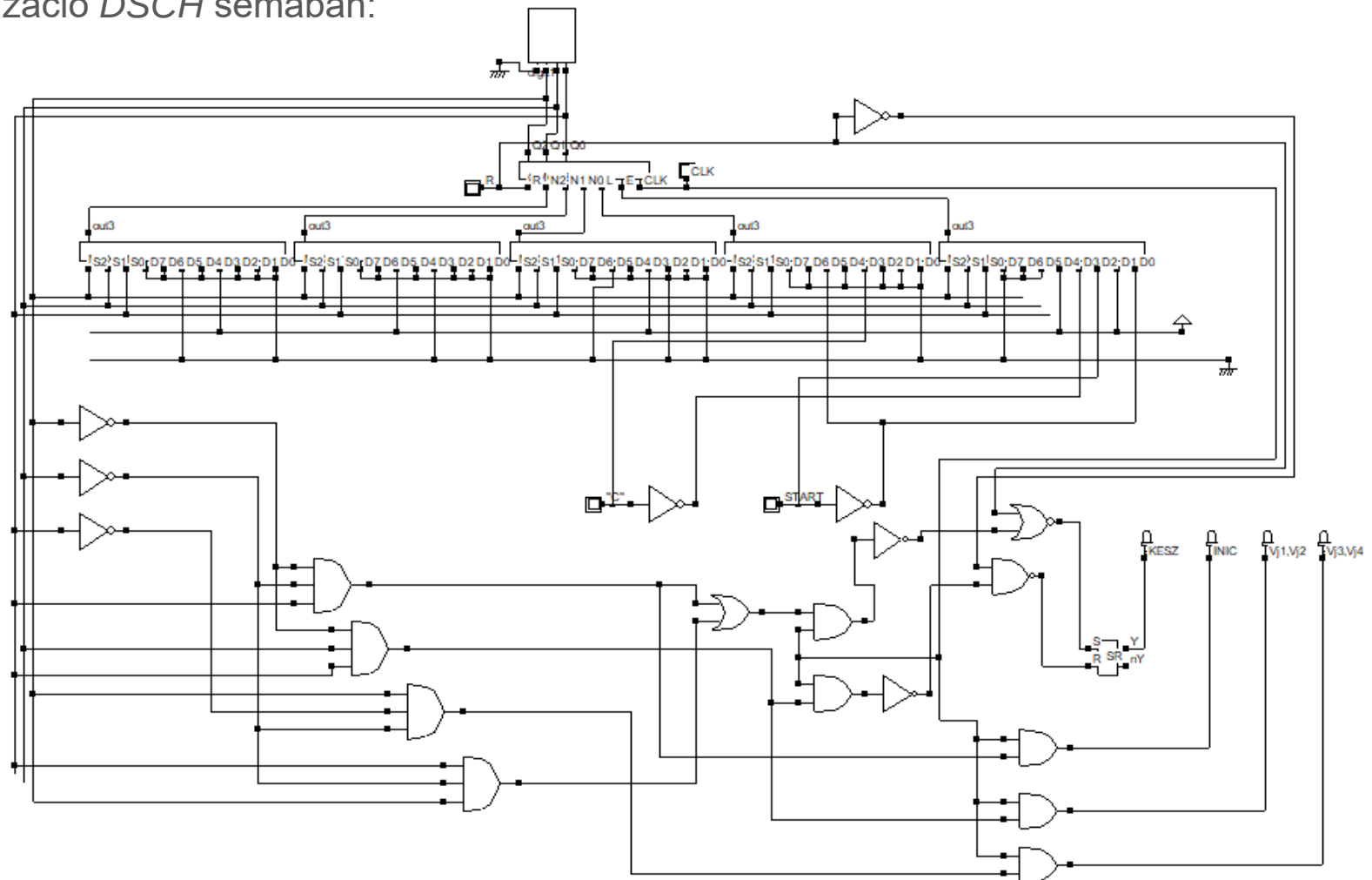


A multiplexerek  
programozása

a vezérlőjelek realizációja

## - 3. mintafeladat -

Realizáció DSCH sémában:



## - 3. mintafeladat -

A feladat megoldásának DSCH sémája  
(„SZH számláló vezérlő előadásmintafeladat 3 dualishoz”):

<http://www.sze.hu/~somi/Digit%e1lis%20h%e1l%3zatok/DSCH%20s%e9m%e1k%20gyakorl%3%20feladatokhoz/>

## Aszinkron sorrendi hálózatok tervezése mintapéldákon keresztül

aszinkron szekvenciális hálózat: aszinkron sorrendi hálózatokban nincs szinkronizáló jel (órajel), azaz a következő állapot ( $n+1$ ) értelmezését nem egy órajel ciklus lezajlása után determináljuk. Ezekben a hálózatokban általánosságban azt feltételezhetjük, hogy a bemenet minden egyes változása egy új állapotot generál.

(megjegyzés: az új állapot ( $n+1$ ) nem feltétlenül különbözik (a szekunder változók és a kimenetek tekintetében) az előzőtől ( $n$ ), ezt mindig a specifikáció szerinti - adott időpontbeli - működési feltételek határozzák meg.)

## Aszinkron sorrendi hálózatok tervezése mintapéldákon keresztül

- 1. mintafeladat -

*Közvetlenül visszacsatolt kombinációs hálózattal tervezzünk olyan egybemenetű ( $X$ ) és egykimenetű ( $Z$ ) aszinkron sorrendi hálózatot, amelynek kimenetén a szint mindannyiszor ellenkezőjére vált, ahányszor a bemenet magas szintről alacsony szintre vált.*

*Bekapcsolás után a hálózat az  $X=0$  bemenetnél  $Z = 0$  kimenetet szolgáltatasson!*

## - 1. mintafeladat -

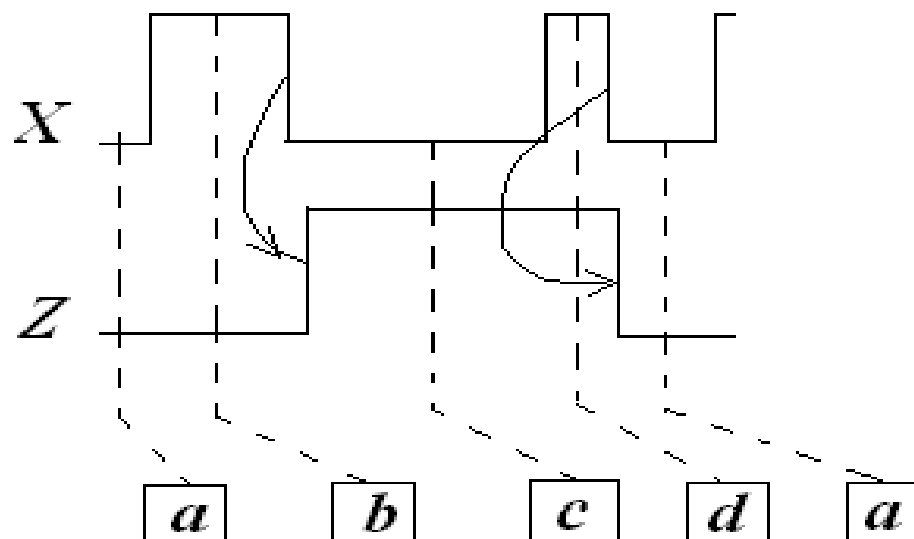
*A feladat megoldásához a következő, általános szisztematikus lépéssort alkalmazzuk:*

- 1. Idődiagram (vagy állapotgráf) felrajzolása.*
- 2. Előzetes szimbolikus állapottábla felvétele.*
- 3. Összevont szimbolikus állapottábla megszerkesztése.*
- 4. Kódolt állapottábla elkészítése a kritikus versenyhelyzet elemzésével.*
- 5. - közvetlen visszacsatolt kombinációs hálózat esetében a szekunder változó(k) és a kimenet(ek) függvényeinek Karnaugh tábla segítségével történő statikus házárdmentesített megadása algebrai alakban;*
  - SR tárolók alkalmazása esetén a vezérlési tábla, valamint a szükséges Karnaugh táblák alapján az SR tároló(k) és a kimenet(ek) statikusan házárdmentesített algebrai függvényalakjainak megadása.*
- 6. Kezdeti állapotról történő gondoskodás.*
- 7. Lényeges házárdok vizsgálata (szükség esetén késleltetések beiktatása).*
- 8. Realizáció logikai kapukkal (és tárolókkal).*

## - 1. mintafeladat -

### Megoldás (Mealy-modell):

1. lépés: idődiagram(ütemdiagram) felrajzolása



## - 1. mintafeladat -

2. lépés: A feladat absztrakt szimbolikus állapotábrájának megszerkesztése

*előzetes szimbolikus állapotábra*

| <b>bem.</b><br><b>akt.áll.</b> | $X=0$               | $X=1$               |
|--------------------------------|---------------------|---------------------|
| $a$                            | $\textcircled{a}/0$ | $b/0$               |
| $b$                            | $c/1$               | $\textcircled{b}/0$ |
| $c$                            | $\textcircled{c}/1$ | $d/1$               |
| $d$                            | $a/0$               | $\textcircled{d}/1$ |

Egy stabil-stabil átmenet ( $a \rightarrow b$ ) az állapotábrán:

| <b>bem.</b><br><b>akt.áll.</b> | $X=0 \rightarrow X=1$               |
|--------------------------------|-------------------------------------|
| $a$                            | $\textcircled{a}/0 \rightarrow b/0$ |
| $b$                            | $c/1 \rightarrow \textcircled{b}/0$ |
| $c$                            | $\textcircled{c}/1 \rightarrow d/1$ |
| $d$                            | $a/0 \rightarrow \textcircled{d}/1$ |

## - 1. mintafeladat -

### 3. lépés: összevont szimbolikus állapottábla megszerkesztése

*előzetes szimbolikus állapottábla*

| <b>bem.</b><br><b>akt.áll.</b> | $X=0$               | $X=1$               |
|--------------------------------|---------------------|---------------------|
| $a$                            | $\textcircled{a}/0$ | $b/0$               |
| $b$                            | $c/1$               | $\textcircled{b}/0$ |
| $c$                            | $\textcircled{c}/1$ | $d/1$               |
| $d$                            | $a/0$               | $\textcircled{d}/1$ |

*Az előzetes szimbolikus állapottáblán – az összevonás feltételeit figyelembe véve – nem tudunk állapot összevonást végezni. A kódolt állapottáblát így e táblázat alapján kell elkészíteni.*

## - 1. mintafeladat -

4. lépés: kódolt állapottábla elkészítése, kritikus versenyhelyzet elemzés

egy lehetséges kód kiosztás:

|       | $Y_1$ | $Y_2$ |
|-------|-------|-------|
| $a :$ | 0     | 0     |
| $b :$ | 0     | 1     |
| $c :$ | 1     | 0     |
| $d :$ | 1     | 1     |



kódolt állapottábla

| kódolt aktuális állapot | kódolt következő állapot/kimenet |                            |
|-------------------------|----------------------------------|----------------------------|
|                         | $X=0$<br>$Y_1^v Y_2^v / Z$       | $X=1$<br>$Y_1^v Y_2^v / Z$ |
| $00$                    | $00/0$                           | $01/0$                     |
| $01$                    | $10/1$                           | $01/0$                     |
| $10$                    | $10/1$                           | $11/1$                     |
| $11$                    | $00/0$                           | $11/1$                     |

## - 1. mintafeladat -

- kritikus versenyhelyzet elemzése a kódolt állapottáblán

| kódolt aktuális<br>állapot<br>$Y_1^v Y_2^v$ | kódolt következő állapot/kimenet |                            |
|---------------------------------------------|----------------------------------|----------------------------|
|                                             | $X=0$<br>$Y_1^v Y_2^v / Z$       | $X=1$<br>$Y_1^v Y_2^v / Z$ |
| 0 0                                         | 0 0 / 0                          | 0 1 / 0                    |
| 0 1                                         | 1 0 / 1                          | 0 1 / 0                    |
| 1 0                                         | 1 0 / 1                          | 1 1 / 1                    |
| 1 1                                         | 0 0 / 0                          | 1 1 / 1                    |

*Egy ideális, '01' → '10' stabil-stabil állapot átmenetet*

## - 1. mintafeladat -

Valóságos állapot-átmenet: kritikus versenyhelyzet az  $Y_1^v Y_2^v = '01' \rightarrow '10'$  állapot átmenet esetében!

| kódolt aktuális állapot<br>$Y_1^v Y_2^v$ | kódolt következő állapot/kimenet |                            |
|------------------------------------------|----------------------------------|----------------------------|
|                                          | $X=0$<br>$Y_1^v Y_2^v / Z$       | $X=1$<br>$Y_1^v Y_2^v / Z$ |
| 00                                       | 00/0                             | 01/0                       |
| 01                                       | 10/1                             | 1. 01/0                    |
| 10                                       | 10/1                             | 2. 11/1                    |
| 11                                       | 00/0                             | 11/1                       |

1.eset:  $Y_1^v Y_2^v = '01' \rightarrow '00' \rightarrow '10'$

- az  $X=1 \rightarrow 0$  hatására az  $Y_1^v Y_2^v = 00$  sorba ugorva a '00' kódú állapotban stabilizálódik a hálózatunk:  
 **$\Rightarrow$  HIBÁS működés!!**

2.eset:  $Y_1^v Y_2^v = '01' \rightarrow '11' \rightarrow '10'$

- az  $X=1 \rightarrow 0$  hatására először az  $Y_1^v Y_2^v = 11$  sorba ugorva a '00' tranzien kód jelenik meg. Innen továbblépve az  $Y_1^v Y_2^v = 00$  sorba ismét a '00' kódú állapotban stabilizálódik a hálózatunk:  
 **$\Rightarrow$  HIBÁS működés!!**

## - 1. mintafeladat -

### **Kritikus versenyhelyzet:**

*amennyiben egy tranziens állapot kódja egynél több szekunder változó értékében különbözik a kiinduló stabil állapot kódjától, a reális hálózaton - az eltérő jelkésleltetési utak miatt - átmenetileg olyan más tranziens állapotok is jelentkezhetnek az  $f_y$  hálózat kimenetén, amelyek stabilizálódhatnak. Ezzel más, a specifikációnak ellentmondó pályára áll az aszinkron hálózat.*

*Az ilyen hibalehetőségeket **kritikus versenyhelyzeteknek** nevezzük.*

## - 1. mintafeladat -

A kritikus versenyhelyzet kiküszöbölése:  
az állapotkódolás megváltoztatása

egy másik lehetséges kódkiosztás:

|       |       |       |
|-------|-------|-------|
|       | $Y_1$ | $Y_2$ |
| $a :$ | 0     | 0     |
| $b :$ | 0     | 1     |
| $c :$ | 1     | 1     |
| $d :$ | 1     | 0     |

$\Rightarrow$

kódolt állapottábla

| kódolt aktuális állapot<br>$Y_1^v Y_2^v$ | kódolt következő állapot/kimenet |                            |
|------------------------------------------|----------------------------------|----------------------------|
|                                          | $X=0$<br>$Y_1^v Y_2^v / Z$       | $X=1$<br>$Y_1^v Y_2^v / Z$ |
| 0 0                                      | 0 0 / 0                          | 0 1 / 0                    |
| 0 1                                      | 1 1 / 1                          | 0 1 / 0                    |
| 1 1                                      | 1 1 / 1                          | 1 0 / 1                    |
| 1 0                                      | 0 0 / 0                          | 1 0 / 1                    |

Nincs kritikus versenyhelyzet!

## - 1. mintafeladat -

5. lépés: a szekunder változók és a kimenet függvényeinek Karnaugh tábla segítségével történő megadása (**hazárdmentesített függvényalak!!**)

**Y1**

|         |   |   |   |   |
|---------|---|---|---|---|
| $Y_1^v$ | 0 | 0 | 1 | 1 |
| $Y_2^v$ | 0 | 1 | 1 | 0 |
| $X$ 0   |   | 1 | 1 |   |
| 1       |   |   | 1 | 1 |

$$Y_1 = Y_2^v \bar{X} + Y_1^v X + Y_1^v Y_2^v$$

**Y2**

|         |   |   |   |   |
|---------|---|---|---|---|
| $Y_1^v$ | 0 | 0 | 1 | 1 |
| $Y_2^v$ | 0 | 1 | 1 | 0 |
| $X$ 0   |   | 1 | 1 |   |
| 1       | 1 | 1 |   |   |

$$Y_2 = \bar{Y}_1^v X + \bar{Y}_1^v Y_2^v + Y_2^v \bar{X}$$

**Z**

|         |   |   |   |   |
|---------|---|---|---|---|
| $Y_1^v$ | 0 | 0 | 1 | 1 |
| $Y_2^v$ | 0 | 1 | 1 | 0 |
| $X$ 0   |   | 1 | 1 |   |
| 1       |   |   | 1 | 1 |

$$Z = Y_1$$

## - 1. mintafeladat -

6. lépés: a kezdeti állapot beállításának ellenőrzése

### Az ellenőrzés menete:

a megoldásként kapott függvényekbe, mint egyenletekbe be kell helyettesíteni a kezdeti állapothoz tartozó ismert és előírt bemeneti értéke(ke)t : ha egyértelműek az eredmények, és teljesül az egyenlőség, akkor nem kell külön kezdő állapot beállító jelről (,RESET') gondoskodni.

Amennyiben azonban a behelyettesítéssel kapott eredmény nem az előírás szerinti kezdő értékeket biztosítja a kimenet(ek) és a szekunder változó(k) tekintetében, illetve nem egyértelmű az eredmény, akkor a következő szabály alapján kell eljárni:



## - 1. mintafeladat -

A kezdeti állapot beállításának általános feltételei(sorrendje):

1. a kezdeti állapot kódját rá kell kényszeríteni az  $f_y$  hálózatra, a visszacsatolástól függetlenül ezeket a bemeneteket. Ezt a helyzetet legalább addig kell fenntartani, amíg az  $f_y$  kimenetein kialakul a kezdeti állapot kódja (illetve, ha **SR** tárolókkal történik a visszacsatolás, akkor azok kimenetén alakul ki ez a kód),
2. rá kell kapcsolni a hálózatra azt a bemeneti értéket (kombinációt), amely a kezdeti állapothoz tartozik,
3. végül meg kell szüntetni a kényszerített visszacsatoló ágat (állapotot), és helyre kell állítani az eredeti visszacsatolást, ezzel a hálózat a kezdeti állapotban stabilizálódik.

## - 1. mintafeladat -

*A kezdeti állapot beállításának ellenőrzése az 1. mintafeladatban:*

$$Y_1 = Y_2^v \overline{X} + Y_1^v X + Y_1^v Y_2^v$$

$$Y_2 = \overline{Y_1^v} X + \overline{Y_1^v} Y_2^v + Y_2^v \overline{X}$$

$$Z = Y_1$$

kezdőállapotban:  
 $X=0$

$\Rightarrow$

$$0 = Y_2^v \cdot 1 + Y_1^v \cdot 0 + Y_1^v Y_2^v$$

$$0 = \overline{Y_1^v} \cdot 0 + \overline{Y_1^v} Y_2^v + Y_2^v \cdot 1$$

$$0 = Y_1$$

bizonytalan!

Konklúzió:

gondoskodni kell a kezdeti állapot beállításáról!  $\rightarrow$  'RESET' (R) jel beiktatása

## - 1. mintafeladat -

A ,RESET' ( $R'$ ) jel beiktatásának elve (a feladat specifikációja alapján):

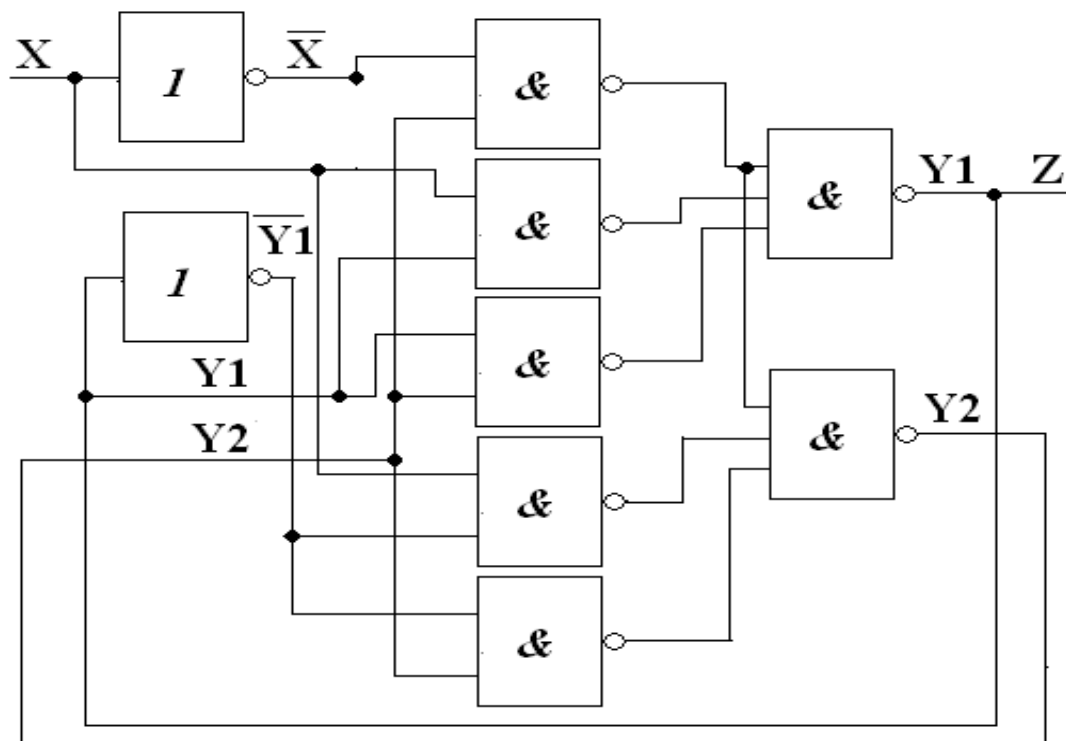
hasítsuk fel a visszacsatolásokat ( $Y_1^v; Y_2^v$ ), és illesszünk be a visszacsatoló körbe 2 darab kétbemenetű logikát. Az egyik bemenetük közös, a kezdeti állapotba kényszerítő ' $R$ ' (reset) jel, a másik bemenetük a visszacsatolandó szekunder változókra ( $Y_1^v; Y_2^v$ ) kapcsolandó. A kimeneteket kapcsoljuk az  $f_y$  hálózat bemeneteire. Mindkét  $R$ -logika az  $R = '1'$  esetben ' $0$ '-át ad tovább, ez pedig a kezdeti állapot kódja. Ha  $R=0$ , a logikák kimenetére a megfelelő szekunder változó kerül, tehát él a visszacsatolás.

## - 1. mintafeladat -

(7. lépés: lényeges hazárdok vizsgálata)

8. lépés: realizáció

*NAND realizáció 'RESET' logika nélkül:*



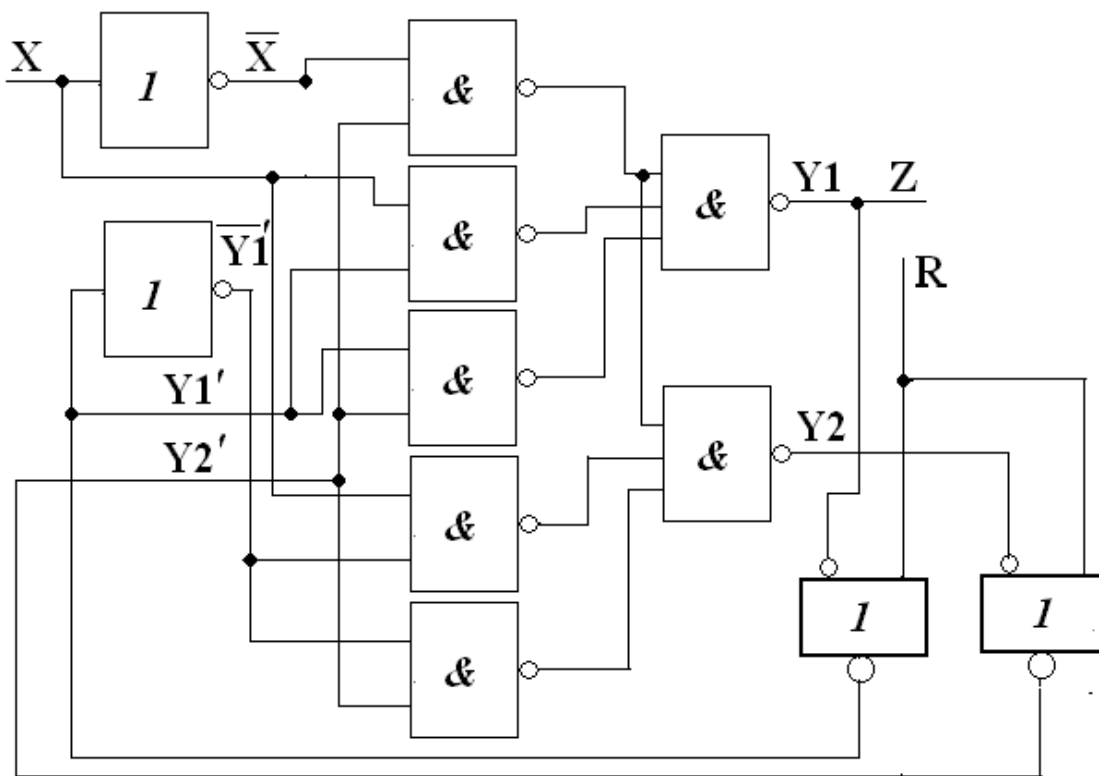
$$Y_1 = Y_2^v \bar{X} + Y_1^v X + Y_1^v Y_2^v$$

$$Y_2 = \bar{Y}_1^v X + \bar{Y}_1^v Y_2^v + Y_2^v \bar{X}$$

$$Z = Y_1$$

## - 1. mintafeladat -

*NAND realizáció 'RESET' logika beiktatásával:*



$$Y_1 = Y_2^v \bar{X} + Y_1^v X + Y_1^v Y_2^v$$

$$Y_2 = \bar{Y}_1^v X + \bar{Y}_1^v Y_2^v + Y_2^v \bar{X}$$

$$Z = Y_1$$

## Aszinkron sorrendi hálózatok tervezése mintapéldákon keresztül

- 2. mintafeladat -

*Tervezzünk kétbemenetű ( $X_1, X_2$ ) ún. „sorrendi ÉS” áramkört, amelynek  $Z$  kimenete akkor és csakis akkor ad magas szintet, ha az  $X_1$  bemenet előbb áll '1'-re, mint az  $X_2$ . Kezdeti állapotban az  $X_1X_2=00$  bemeneti bitkombináció esetén  $Z=0$ .*

*A tervezést végezzük el a következő állapotot előállító hálózat közvetlen visszacsatolásával, és **SR** tárolókkal történő visszacsatolással is !*

## - 2. mintafeladat -

### Megoldás 1.: közvetlen visszacsatolás alkalmazása (Mealy-modell szerint)

(1. lépés: idődiagram vagy állapotgráf felrajzolása – elhagyható)

1. lépés: előzetes szimbolikus állapottábla felvétele a specifikáció alapján

| aktuális<br>állapot | következő állapot/kimenet |             |             |             |
|---------------------|---------------------------|-------------|-------------|-------------|
|                     | $X_1 X_2$                 |             |             |             |
|                     | 00                        | 01          | 10          | 11          |
| <i>a</i>            | ⓐ/0                       | <i>b</i> /0 | <i>c</i> /0 | -/-         |
| <i>b</i>            | <i>a</i> /0               | ⓑ/0         | -/-         | <i>d</i> /0 |
| <i>c</i>            | <i>a</i> /0               | -/-         | ⓒ/0         | <i>e</i> /1 |
| <i>d</i>            | -/-                       | <i>b</i> /0 | <i>c</i> /0 | ⓓ/0         |
| <i>e</i>            | -/-                       | <i>b</i> /0 | <i>c</i> /0 | ⓔ/1         |

## - 2. mintafeladat -

2. lépés: összevont szimbolikus állapottábla megszerkesztése

*előzetes szimbolikus állapottábla*

| aktuális<br>állapot | következő állapot/kimenet |                     |                     |                     |
|---------------------|---------------------------|---------------------|---------------------|---------------------|
|                     | $X_1 X_2$                 |                     |                     |                     |
|                     | 00                        | 01                  | 10                  | 11                  |
| <i>a</i>            | $\textcircled{a}/0$       | <i>b/0</i>          | <i>c/0</i>          | <i>-/-</i>          |
| <i>b</i>            | <i>a/0</i>                | $\textcircled{b}/0$ | <i>-/-</i>          | <i>d/0</i>          |
| <i>c</i>            | <i>a/0</i>                | <i>-/-</i>          | $\textcircled{c}/0$ | <i>e/1</i>          |
| <i>d</i>            | <i>-/-</i>                | <i>b/0</i>          | <i>c/0</i>          | $\textcircled{d}/0$ |
| <i>e</i>            | <i>-/-</i>                | <i>b/0</i>          | <i>c/0</i>          | $\textcircled{e}/1$ |

összevonható állapotpárok:

*ab, ad, bd, ce*

összevont állapotok osztályai:

*abd, ce*

$abd \rightarrow s_1$

$ce \rightarrow s_2$

## - 2. mintafeladat -

$abd \rightarrow s_1$

$ce \rightarrow s_2$

előzetes szimbolikus állapottábla

| aktuális állapot | következő állapot/kimenet |                     |                     |                     |
|------------------|---------------------------|---------------------|---------------------|---------------------|
|                  | $X_1 X_2$                 |                     |                     |                     |
|                  | 00                        | 01                  | 10                  | 11                  |
| $a$              | $\textcircled{a}/0$       | $b/0$               | $c/0$               | $-/-$               |
| $b$              | $a/0$                     | $\textcircled{b}/0$ | $-/-$               | $d/0$               |
| $c$              | $a/0$                     | $-/-$               | $\textcircled{c}/0$ | $e/1$               |
| $d$              | $-/-$                     | $b/0$               | $c/0$               | $\textcircled{d}/0$ |
| $e$              | $-/-$                     | $b/0$               | $c/0$               | $\textcircled{e}/1$ |



összevont szimbolikus állapottábla

| aktuális állapot, | következő állapot/kimenet |                       |                       |                       |
|-------------------|---------------------------|-----------------------|-----------------------|-----------------------|
|                   | $X_1 X_2$                 |                       |                       |                       |
|                   | 00                        | 01                    | 10                    | 11                    |
| $s_1$             | $\textcircled{s_1}/0$     | $\textcircled{s_1}/0$ | $s_2/0$               | $\textcircled{s_1}/0$ |
| $s_2$             | $s_1/0$                   | $s_1/0$               | $\textcircled{s_2}/0$ | $\textcircled{s_2}/1$ |

## - 2. mintafeladat -

3. lépés: kódolt állapottábla elkészítése (kritikus versenyhelyzet elemzés: nem szükséges)

egy lehetséges kód kiosztás:

$Y$   
 $s_1 : 0$   
 $s_2 : 1$

$\Rightarrow$

kódolt állapottábla

| aktuális<br>állapot,<br>$Y$ | következő állapot/kimenet<br>$X_1X_2$ |                     |                     |                     |
|-----------------------------|---------------------------------------|---------------------|---------------------|---------------------|
|                             | $00$                                  | $01$                | $10$                | $11$                |
| $0$                         | $\textcircled{0}/0$                   | $\textcircled{0}/0$ | $1/0$               | $\textcircled{0}/0$ |
| $1$                         | $0/0$                                 | $0/0$               | $\textcircled{1}/0$ | $\textcircled{1}/1$ |

## - 2. mintafeladat -

4. lépés: a szekunder változó és a kimenet függvényeinek Karnaugh tábla segítségével történő megadása

**Y**

|         |   |   |   |   |
|---------|---|---|---|---|
| $X1$    | 0 | 0 | 1 | 1 |
| $X2$    | 0 | 1 | 1 | 0 |
| $Y^v$ 0 |   |   |   | 1 |
| 1       |   |   | 1 | 1 |

$$Y = X_1 \overline{X_2} + X_1 Y^v$$

**Z**

|         |   |   |   |   |
|---------|---|---|---|---|
| $X1$    | 0 | 0 | 1 | 1 |
| $X2$    | 0 | 1 | 1 | 0 |
| $Y^v$ 0 |   |   |   |   |
| 1       |   |   | 1 |   |

$$Z = X_1 X_2 Y^v$$

## - 2. mintafeladat -

5.lépés: a kezdeti állapot beállításának ellenőrzése

$$\begin{array}{llll} Y = X_1 \overline{X_2} + X_1 Y^v & \text{kezdőállapotban:} & 0(Y) = 0 \cdot 1 + 0 \cdot Y^v & \checkmark \text{😊} \\ & X_1 X_2 = 00 & & \\ Z = X_1 X_2 Y^v & \Rightarrow & 0(Z) = 0 \cdot 0 \cdot Y^v & \checkmark \text{😊} \end{array}$$

### Konklúzió:

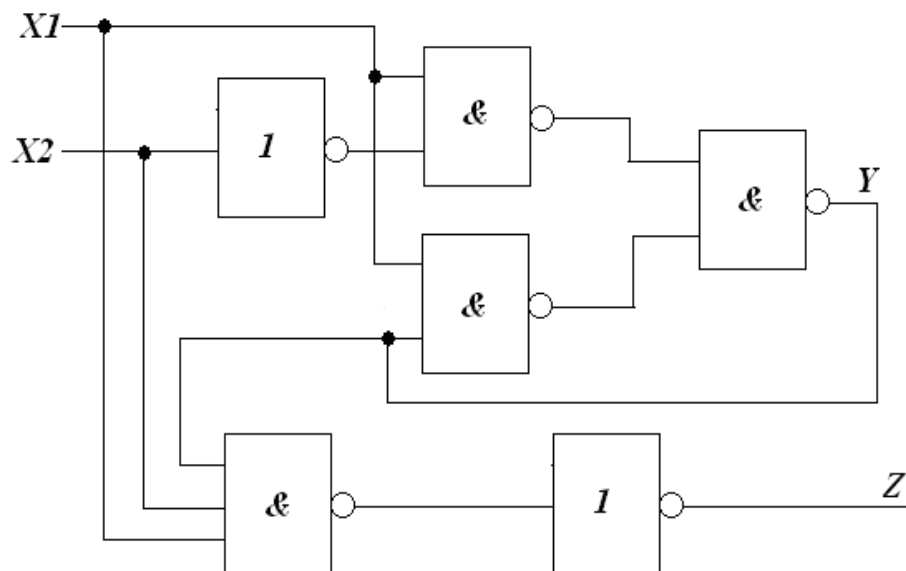
nem kell gondoskodni a kezdeti állapot 'RESET' jellel történő beállításáról

## - 2. mintafeladat -

(6. lépés: lényeges hazárdok vizsgálata)

7. lépés: realizáció

*NAND realizáció ('RESET' logika nélkül):*



$$Y = X_1 \overline{X_2} + X_1 Y^v$$

$$Z = X_1 X_2 Y^v$$

## - 2. mintafeladat -

### Megoldás 2.: SR tárolóval történő visszacsatolás alkalmazása (Mealy-modell szerint)

1.-3. lépés: megegyezik az előző feladatmegoldás lépéseivel

4. lépés: az alkalmazott SR tároló vezérlésének felírása a kódolt állapottábla alapján

**Y:** az aktuális állapotot reprezentáló szekunder változó

| aktuális állapot,<br>$Y$ | következő állapot/kimenet<br>$X_1 X_2$ |     |     |     |
|--------------------------|----------------------------------------|-----|-----|-----|
|                          | 00                                     | 01  | 10  | 11  |
| 0                        | 0/0                                    | 0/0 | 1/0 | 0/0 |
| 1                        | 0/0                                    | 0/0 | 1/0 | 1/1 |

## - 2. mintafeladat -

Az SR tároló  
saját vezérlési  
táblája

| $Y^V \rightarrow Y$ |   | S | R |
|---------------------|---|---|---|
| 0                   | 0 | 0 | — |
| 0                   | 1 | 1 | 0 |
| 1                   | 0 | 0 | 1 |
| 1                   | 1 | — | 0 |

kódolt állapottábla

| aktuális<br>állapot,<br>$Y$ | következő állapot/kimenet |     |     |     |
|-----------------------------|---------------------------|-----|-----|-----|
|                             | $X_1 X_2$                 |     |     |     |
|                             | 00                        | 01  | 10  | 11  |
| 0                           | 0/0                       | 0/0 | 1/0 | 0/0 |
| 1                           | 0/0                       | 0/0 | 1/0 | 1/1 |



vezérlési tábla

| kódolt<br>aktuális<br>állapot<br>$Y$ | kódolt következő állapot |     |     |     |
|--------------------------------------|--------------------------|-----|-----|-----|
|                                      | $X_1 X_2$                |     |     |     |
|                                      | 00                       | 01  | 10  | 11  |
|                                      | S R                      | S R | S R | S R |
| 0                                    | 0 -                      | 0 - | 1 0 | 0 - |
| 1                                    | 0 1                      | 0 1 | - 0 | - 0 |



## - 2. mintafeladat -

5. lépés: az SR tároló és a kimenet függvényeinek Karnaugh tábla segítségével történő megadása

**S**

|         |   |   |   |   |
|---------|---|---|---|---|
| $X_1$   | 0 | 0 | 1 | 1 |
| $X_2$   | 0 | 1 | 1 | 0 |
| $Y^v$ 0 |   |   |   | 1 |
| 1       |   |   | - | - |

**R**

|         |   |   |   |   |
|---------|---|---|---|---|
| $X_1$   | 0 | 0 | 1 | 1 |
| $X_2$   | 0 | 1 | 1 | 0 |
| $Y^v$ 0 | - | - | - |   |
| 1       | 1 | 1 |   |   |

$$S = X_1 \overline{X_2}$$

$$R = \overline{X_1}$$

$$Z = X_1 X_2 Y^v$$

**Z**

|         |   |   |   |   |
|---------|---|---|---|---|
| $X_1$   | 0 | 0 | 1 | 1 |
| $X_2$   | 0 | 1 | 1 | 0 |
| $Y^v$ 0 |   |   |   |   |
| 1       |   |   | 1 |   |

## - 2. mintafeladat -

6.lépés: a kezdeti állapot beállításának ellenőrzése

kezdeti állapot:  $Y=0!$   $\rightarrow$  szükséges SR vezérlés:  $\left\{ \begin{matrix} S=0 \\ R=1 \end{matrix} \right\}$

$\downarrow$

$$S = X_1 \overline{X_2}$$

$$R = \overline{X_1}$$

$$Z = X_1 X_2 Y^v$$

kezdőállapotban:

$$X_1 X_2 = 00$$

$\Rightarrow$

$$0(S) = 0 \cdot 1$$

$$1(R) = 1$$

$$0(Z) = 0 \cdot 0 \cdot Y^v$$

✓ 😊

✓ 😊

✓ 😊

Konklúzió:

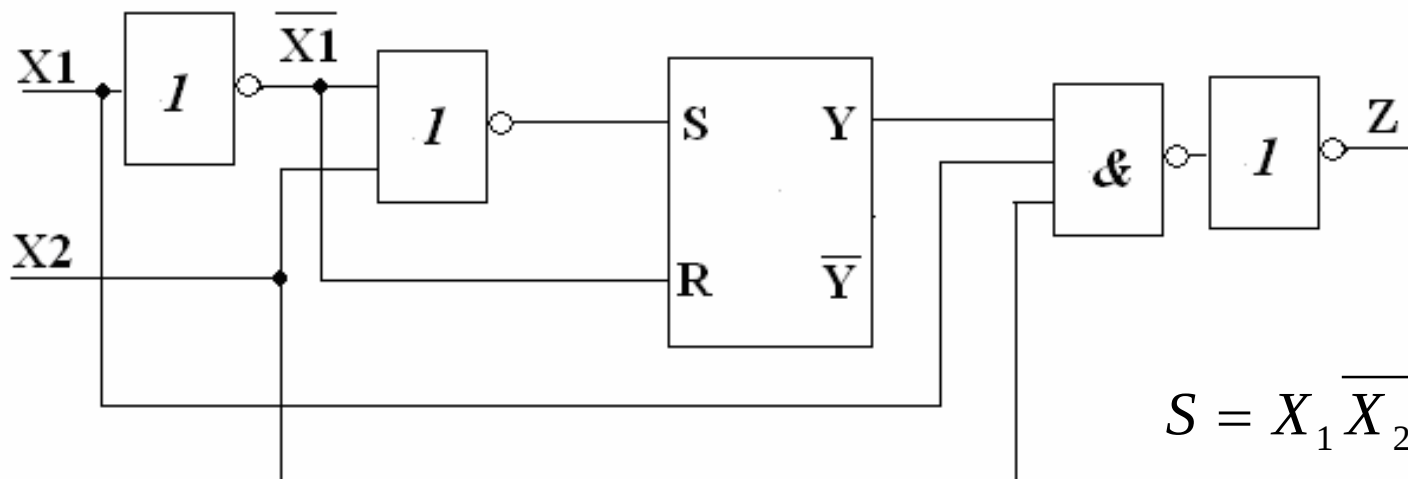
nem kell gondoskodni a kezdeti állapot 'RESET' jellel történő beállításáról

## - 2. mintafeladat -

(7. lépés: lényeges hazárdok vizsgálata)

8. lépés: realizáció

*NAND realizáció ('RESET' logika nélkül):*



$$S = X_1 \overline{X_2}$$

$$R = \overline{X_1}$$

$$Z = X_1 X_2 Y^v$$

## Lényeges hazardok aszinkron hálózatokban

Az eddigi aszinkron hálózat tervezési példáink megoldása során csak a szekunder változók versengése miatt kialakuló hibákkal, és azok kiküszöbölésével foglalkoztunk. Ez csak akkor tekinthető korrekt eljárásnak, ha garantálni tudjuk azt, hogy a bemeneti jelek változása okozta események a szekunder változók értékeinek megváltozása kezdete előtt már lezajlanak. Ez a feltételezésünk abban is megnyilvánul, hogy amikor az állapottáblán követjük az aszinkron hálózat működését, egyik oszlopról a másikra térünk át, és csak ezután vizsgáljuk a tranzienseket. A valóságban ez a feltételezés nem mindig jogos. A szekunder változók és egyik bemeneti változó kritikus versenyhelyzete úgynevezett lényeges hazard veszélyével jár. Ennek kiküszöbölése időkésleltetési manipulációkat igényel.

## Lényeges hazárdok aszinkron hálózatokban

### Példa:

aszinkron szekvenciális  
hálózat (1.mintapélda)  
működésének analízálása  
az állapottáblán:

| kódolt aktuális<br>állapot<br>$Y_1^v Y_2^v$ | kódolt következő állapot/kimenet |                            |
|---------------------------------------------|----------------------------------|----------------------------|
|                                             | $X=0$<br>$Y_1^v Y_2^v / Z$       | $X=1$<br>$Y_1^v Y_2^v / Z$ |
| 0 0                                         | 0 0 / 0                          | 0 1 / 0                    |
| 0 1                                         | 1 0 / 1                          | 0 1 / 0                    |
| 1 0                                         | 1 0 / 1                          | 1 1 / 1                    |
| 1 1                                         | 0 0 / 0                          | 1 1 / 1                    |

## Sorrendi hálózatok kezdeti állapotának beállítási lehetőségei

- Szinkron sorrendi hálózatok kezdeti állapotának beállítása:
  - beállítás a 'preset' (Pr) és a 'clear' (Clr) segédbemenetek felhasználásával
  - beállítás az  $f_y$  hálózat kiegészítésével
  - (beállítás a szekunder változók aktuális állapotának módosításával)
  
- Aszinkron sorrendi hálózatok kezdeti állapotának beállítása:
  - közvetlenül visszacsatolt kombinációs hálózattal megvalósított aszinkron szekvenciális hálózat kezdeti állapotának beállítása
  - SR tárolókkal megvalósított aszinkron szekvenciális hálózat kezdeti állapotának beállítása
  - (beállítás a szekunder változók aktuális állapotának módosításával)

- *beállítás a ,preset' (Pr) és a ,clear' (Clr) segédbemenetek felhasználásával*



## Szinkron sorrendi hálózatok kezdeti állapotának beállítása (II.)

- *beállítás az  $f_y$  hálózat kiegészítésével D-MS flip-flop esetében:  
kiegészítés RESET segédbemenettel*

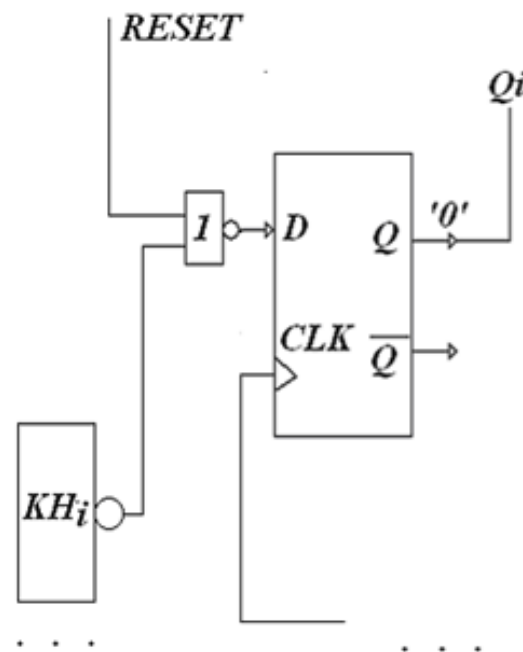
a) szükséges kezdeti állapot: '0' ( $Q=0$ )

$(D \rightarrow D_i)$

| $D_i$ | RESET | $D_i'$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

$$D_i' = D_i \overline{RESET} = \overline{\overline{D_i} + RESET}$$

(Ez a módszer minden esetben biztosítja a kezdeti állapot beállítását a szekunder változók és a bemenetek aktuális állapotától függetlenül!)



## Szinkron sorrendi hálózatok kezdeti állapotának beállítása (II.)

- *beállítás az  $f_y$  hálózat kiegészítésével D-MS flip-flop esetében:  
kiegészítés RESET segédbemenettel*

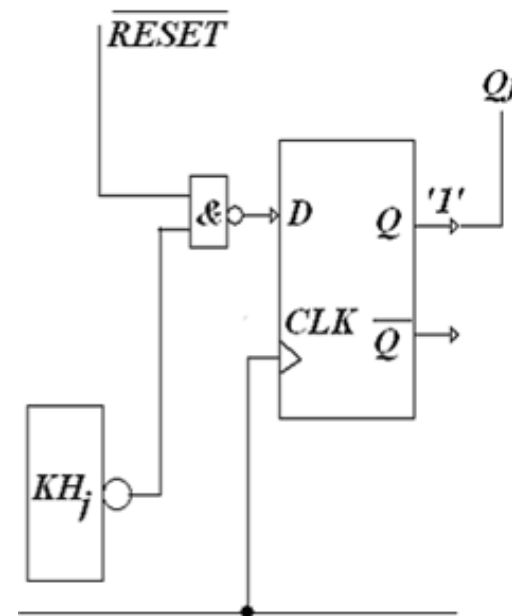
b) szükséges kezdeti állapot: '1' (Q=1)

$(D \rightarrow D_j)$

| $D_j$ | RESET | $D'_j$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

$$D'_j = \overline{\overline{D_j} \overline{RESET}}$$

(Ez a módszer minden esetben biztosítja a kezdeti állapot beállítását a szekunder változók és a bemenetek aktuális állapotától függetlenül!)



## Szinkron sorrendi hálózatok kezdeti állapotának beállítása (II.)

- *beállítás az  $f_y$  hálózat kiegészítésével JK-MS flip-flop esetében:  
kiegészítés RESET segédbemenettel*

a) szükséges kezdeti állapot: „0” (Q=0)

$(J \rightarrow J_i)$

| $J_i$ | RESET | $J'_i$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

$$J'_i = \overline{\overline{J_i} + RESET}$$

$(K \rightarrow K_i)$

| $K_i$ | RESET | $K'_i$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

$$K'_i = \overline{\overline{K_i} RESET}$$

(Ez a módszer minden esetben biztosítja a kezdeti állapot beállítását a szekunder változók és a bemenetek aktuális állapotától függetlenül!)

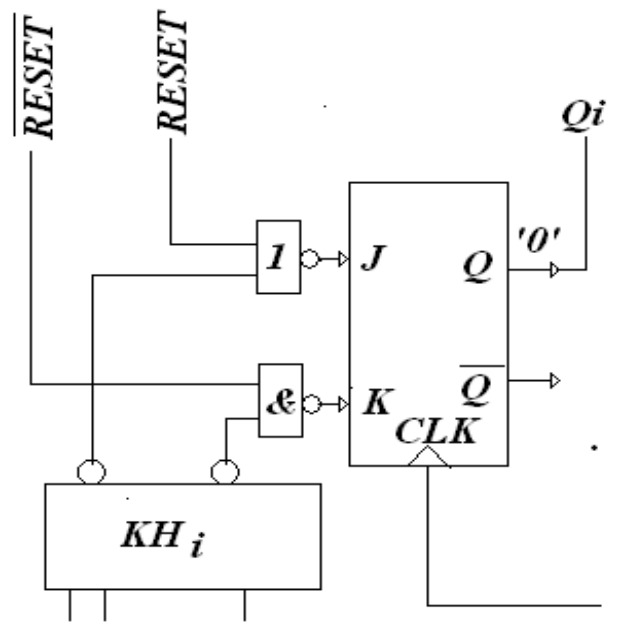
## Szinkron sorrendi hálózatok kezdeti állapotának beállítása (II.)

- *beállítás az  $f_y$  hálózat kiegészítésével JK-MS flip-flop esetében:  
kiegészítés RESET segédbemenettel*

a) szükséges kezdeti állapot: „0” (Q=0)

$$J'_i = \overline{\overline{J_i} + RESET}$$

$$K'_i = \overline{\overline{\overline{K_i}} RESET}$$



## Szinkron sorrendi hálózatok kezdeti állapotának beállítása (II.)

- *beállítás az  $f_y$  hálózat kiegészítésével JK-MS flip-flop esetében:  
kiegészítés RESET segédbemenettel*

a) szükséges kezdeti állapot: „1” (Q=1)

$(J \rightarrow J_j)$

| $J_j$ | RESET | $J'_j$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

$$J'_j = \overline{\overline{J_j} \overline{RESET}}$$

$(K \rightarrow K_j)$

| $K_j$ | RESET | $K'_j$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

$$K'_j = \overline{\overline{K_j} + RESET}$$

(Ez a módszer minden esetben biztosítja a kezdeti állapot beállítását a szekunder változók és a bemenetek aktuális állapotától függetlenül!)

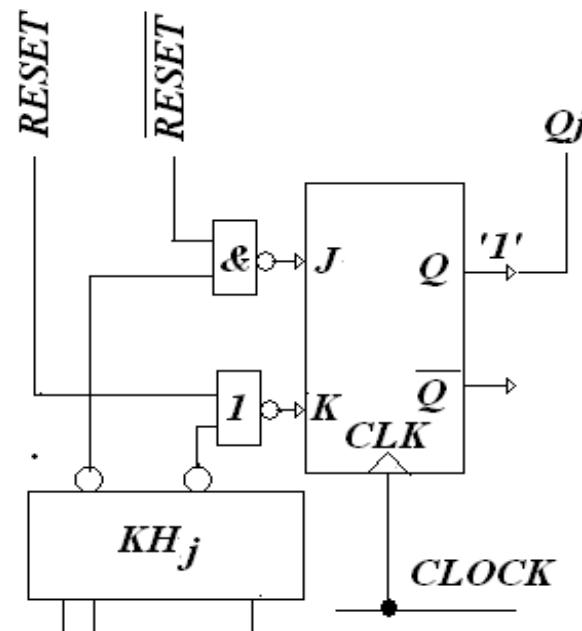
## Szinkron sorrendi hálózatok kezdeti állapotának beállítása (II.)

- *beállítás az  $f_y$  hálózat kiegészítésével JK-MS flip-flop esetében:  
kiegészítés RESET segédbemenettel*

a) szükséges kezdeti állapot: ,1' (Q=1)

$$J'_j = \overline{\overline{\overline{J_j}} \overline{RESET}}$$

$$K'_j = \overline{\overline{\overline{K_j}} + RESET}$$



## Szinkron sorrendi hálózatok kezdeti állapotának beállítása (III.)

- beállítás a szekunder változók aktuális állapotának módosításával (elv)*

a) legyen:  $Q \rightarrow 0$  !

| $Q_i$ | RESET | $Q_i'$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

b) legyen:  $Q \rightarrow 1$  !

| $Q_j$ | RESET | $Q_j'$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

(Ez a módszer egyszerűbb, de nem biztosítja minden esetben a bemenetektől függetlenül a kezdeti állapot beállítását.

pl.:

$$J1 = Q_2^n X1 \overline{X2}$$

$$K1 = 1$$

$$Z = Q_1^n$$

$$J2 = \overline{X1} X2$$

$$K2 = \overline{Q_1^n} \overline{Q_2^n} \overline{X1} X2$$

$Q_1^n$  és  $Q_2^n$  alacsony szintje nem garantálja mindkét J alacsony, és mindkét K magas szintjét!

## Aszinkron sorrendi hálózatok kezdeti állapotának beállítása (I.)

- közvetlenül visszacsatolt kombinációs hálózattal megvalósított aszinkron szekvenciális hálózat kezdeti állapotának beállítása a szekunder változók módosításával*

a) szükséges kezdeti állapot: ,0' (Y=0)

$(Y \rightarrow Y_i)$

| $Y_i$ | RESET | $Y_i'$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

(Ez a módszer csak a bemenetekre megadott kezdeti bemeneti kombinációval együtt biztosítja a stabil kezdeti állapot beállítását!)

b) szükséges kezdeti állapot: ,1' (Y=1)

$(Y \rightarrow Y_j)$

| $Y_j$ | RESET | $Y_j'$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

## Aszinkron sorrendi hálózatok kezdeti állapotának beállítása (II.)

- SR tárolókkal visszacsatolt aszinkron szekvenciális hálózatok kezdeti állapotának beállítása az  $f_y$  hálózat  $S$  és  $R$  kimeneteinek kiegészítésével*

a) szükséges kezdeti állapot: „0” ( $Y=0$ )

$(S \rightarrow S_i)$

| $S_i$ | RESET | $S'_i$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

$$S'_i = \overline{\overline{S_i} + RESET}$$

$(R \rightarrow R_i)$

| $R_i$ | RESET | $R'_i$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

$$R'_i = \overline{\overline{R_i} \overline{RESET}}$$

(Ez nem biztosítja minden esetben a bemenetektől függetlenül a stabil kezdeti állapot beállítását!

pl.:

$$S_1 = Y_2^V \overline{X} \quad R_1 = \overline{Y_2^V} \overline{X}$$

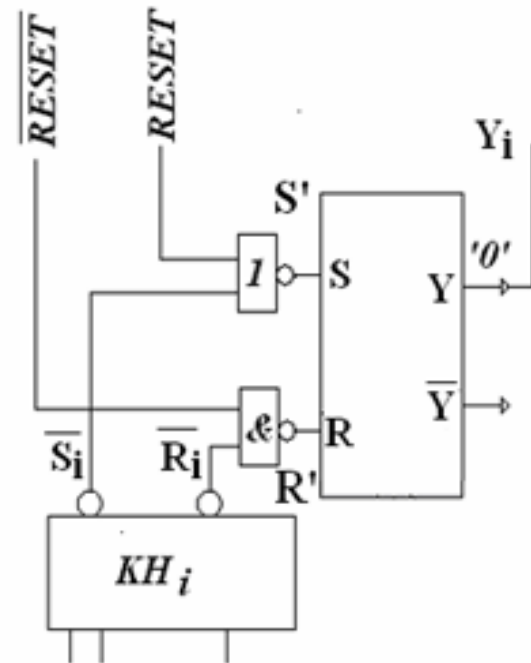
$$S_2 = \overline{Y_1^V} X \quad R_2 = Y_1^V X$$

$$Z = R_2$$

- *SR tárolókkal visszacsatolt aszinkron szekvenciális hálózatok kezdeti állapotának beállítása az  $f_y$  hálózat  $S$  és  $R$  kimeneteinek kiegészítésével*

$$S'_i = \overline{\overline{S_i}} + RESET$$

$$R'_i = \overline{\overline{R_i} RESET}$$



## Aszinkron sorrendi hálózatok kezdeti állapotának beállítása (II.)

- SR tárolókkal visszacsatolt aszinkron szekvenciális hálózatok kezdeti állapotának beállítása az  $f_y$  hálózat  $S$  és  $R$  kimeneteinek kiegészítésével*

a) szükséges kezdeti állapot: „1” ( $Y=1$ )

$(S \rightarrow S_j)$

| $S_j$ | RESET | $S'_j$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

$$S'_j = \overline{\overline{S_j}} \overline{\overline{RESET}}$$

$(R \rightarrow R_j)$

| $R_j$ | RESET | $R'_j$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

$$R'_j = \overline{\overline{R_j}} + \overline{\overline{RESET}}$$

(Ez nem biztosítja minden esetben a bemenetektől függetlenül a stabil kezdeti állapot beállítását!

pl.:

$$S_1 = Y_2^V \overline{X} \quad R_1 = \overline{Y_2^V} \overline{X}$$

$$S_2 = \overline{Y_1^V} X \quad R_2 = Y_1^V X$$

$$Z = R_2$$

- *SR tárolókkal visszacsatolt aszinkron szekvenciális hálózatok kezdeti állapotának beállítása az  $f_y$  hálózat  $S$  és  $R$  kimeneteinek kiegészítésével*

$$S'_j = \overline{\overline{S_j}} \overline{\overline{RESET}}$$

## Aszinkron sorrendi hálózatok kezdeti állapotának beállítása (III.)

- SR tárolókkal visszacsatolt aszinkron szekvenciális hálózat kezdeti állapotának beállítása a szekunder változók módosításával (elv)*

a) szükséges kezdeti állapot: ,0' (Y=0)

$(Y \rightarrow Y_i)$

| $Y_i$ | RESET | $Y_i'$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

(Ez a módszer a bemenetekre megadott kezdeti bemeneti kombinációval együtt sem biztosítja mindig a kezdeti állapot beállítását! /lásd előző példa/.)

b) szükséges kezdeti állapot: ,1' (Y=1)

$(Y \rightarrow Y_j)$

| $Y_j$ | RESET | $Y_j'$ |
|-------|-------|--------|
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

## Állapot összevonási módszerek

- Állapot összevonási módszerek teljesen specifikált szekvenciális hálózatok esetében
  - állapot összevonás az ekvivalencia-típusú reláció alapján
  - az összevonás lépései, a lépcsős tábla alkalmazása az ekvivalencia-típusú reláció alapján
  
- Állapot összevonási módszerek nem teljesen specifikált szekvenciális hálózatok esetében
  - állapot összevonás a kompatibilitás-típusú reláció alapján
  - az összevonás lépései, a lépcsős tábla alkalmazása a kompatibilitás-típusú reláció alapján

## Állapot összevonás teljesen specifikált előzetes szimbolikus állapottáblán

### Emlékeztető

(...korábban...)

Az állapot összevonás feltételei:

*az előzetes szimbolikus állapottábla két állapotát nem kell megkülönböztetni, ezért azok összevonhatók,*

- ha bemeneti kombinációként egyeznek a hozzájuk rendelt kimeneti kombinációk,*
- és bemenő kombinációként ugyanarra a következő állapotra vezetnek.*

## Állapot összevonás teljesen specifikált előzetes szimbolikus állapottáblán

### Az összevonhatóság (szükséges és elégséges) feltételei:

*Két állapot a teljesen specifikált hálózat állapottábláján nem megkülönböztethető (NMK), ha a két állapotból elindulva bármely bemeneti sorozatra ugyanazt a kimeneti sorozatot látjuk.*

*Ebből a magától értetődő definícióból kiindulva bizonyítható, hogy két állapot összevonható, ha a két állapotból bármely bemeneti kombinációra adott kimeneti kombinációk megegyeznek, és nem megkülönböztethető állapotokra vezetnek.*

## A „nem megkülönböztethetőség”(NMK), mint reláció

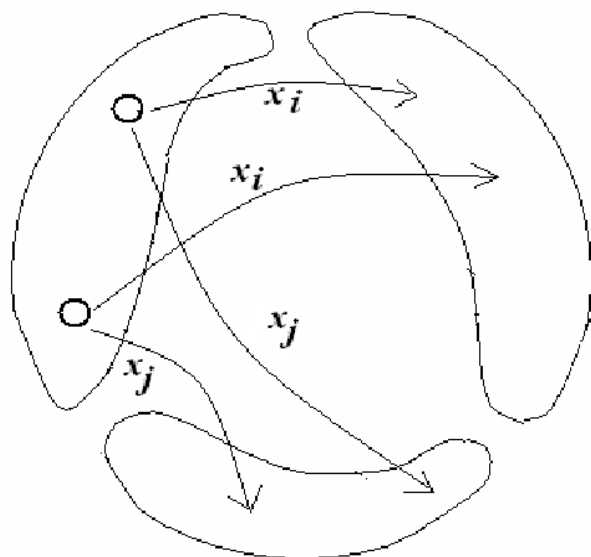
### Tulajdonságok:

1. *reflexív*
2. *szimmetrikus*
3. *tranzitív*

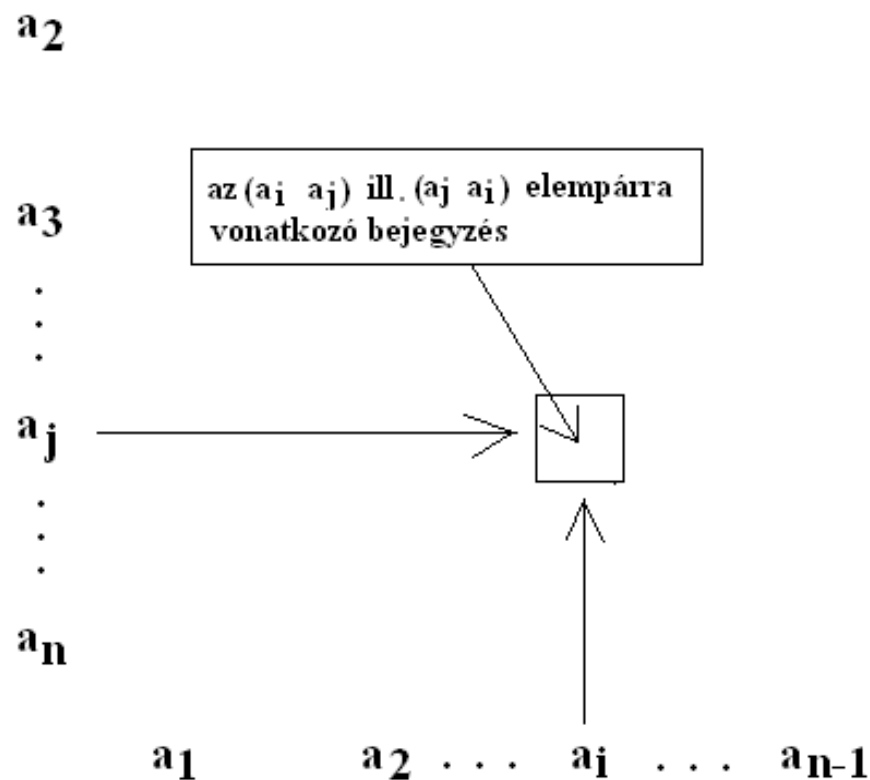
1. a *reflexivitás* az a trivialis, hogy egy szimbolikus állapot saját magától nem különböztethető meg, azaz  $a \equiv a$ .
2. *szimmetrikusak* azok a bináris relációk, amelyekre igaz, hogy amennyiben  $a \equiv b$ , akkor bizonyosan fennáll a  $b \equiv a$  reláció is.
3. *tranzitív* relációk esetén igaz, hogy amennyiben  $a \equiv b$  és  $b \equiv c$ , akkor teljesül az  $a \equiv c$  is.

Az ilyen relációkat **ekvivalencia-típusú reláció**knak nevezzük.

## Összevonható állapotok szemléltetése és a lépcsős tábla



*Diszjunkt részhalmazokra  
bontás*



## Jelölések a lépcsős táblán

- $a \equiv b$  : az  $a$  és a  $b$  állapotok ekvivalensek
- $a \neq b$  : az  $a$  és a  $b$  állapotok antivalensek
- *feltételes ekvivalencia*: a feltételes ekvivalenciát magával a feltétellel jelöljük. Például, ha a jelölés a következő felsorolás:  $(ab, cd \dots)$  akkor az a két állapot, amelyekre ez vonatkozik, feltételesen ekvivalensek, azaz csak akkor ekvivalensek, ha  $a \equiv b$  és  $c \equiv d$ .

## Mintapélda megoldása lépcsős táblán (1)

Példa:

*előzetes szimbolikus állapottábla*

| aktuális állapot | következő állapot/kimenet |            |
|------------------|---------------------------|------------|
|                  | $X=0$                     | $X=1$      |
| <b>a</b>         | <b>c/0</b>                | <b>b/1</b> |
| <b>b</b>         | <b>d/1</b>                | <b>e/0</b> |
| <b>c</b>         | <b>a/0</b>                | <b>d/1</b> |
| <b>d</b>         | <b>b/1</b>                | <b>e/0</b> |
| <b>e</b>         | <b>e/0</b>                | <b>a/1</b> |



|          |                        |                   |                        |                   |
|----------|------------------------|-------------------|------------------------|-------------------|
| <b>b</b> | $\langle \rangle$      |                   |                        |                   |
| <b>c</b> | <b>bd</b>              | $\langle \rangle$ |                        |                   |
| <b>d</b> | $\langle \rangle$      | $\equiv$          | $\langle \rangle$      |                   |
| <b>e</b> | <b>ab</b><br><b>ce</b> | $\langle \rangle$ | <b>ad</b><br><b>ae</b> | $\langle \rangle$ |
|          | <b>a</b>               | <b>b</b>          | <b>c</b>               | <b>d</b>          |

*1.generációs lépcsős tábla*

## Mintapélda megoldása lépcsős táblán (2)

|          |                        |          |                        |          |
|----------|------------------------|----------|------------------------|----------|
| <b>b</b> | <>                     |          |                        |          |
| <b>c</b> | <b>bd</b>              | <>       |                        |          |
| <b>d</b> | <>                     | ≡        | <>                     |          |
| <b>e</b> | <b>ab</b><br><b>ce</b> | <>       | <b>ad</b><br><b>ae</b> | <>       |
|          | <b>a</b>               | <b>b</b> | <b>c</b>               | <b>d</b> |

1.generációs lépcsős tábla



|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| <b>b</b> | <>       |          |          |          |
| <b>c</b> | ≡        | <>       |          |          |
| <b>d</b> | <>       | ≡        | <>       |          |
| <b>e</b> | <>       | <>       | <>       | <>       |
|          | <b>a</b> | <b>b</b> | <b>c</b> | <b>d</b> |

2.generációs lépcsős tábla

## Mintapélda megoldása lépcsős táblán (3)

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| <b>b</b> | < >      |          |          |          |
| <b>c</b> | ≡        | < >      |          |          |
| <b>d</b> | < >      | ≡        | < >      |          |
| <b>e</b> | < >      | < >      | < >      | < >      |
|          | <b>a</b> | <b>b</b> | <b>c</b> | <b>d</b> |

$$\mathbf{a} \equiv \mathbf{c}$$

$$\mathbf{b} \equiv \mathbf{d}$$

**e**

( **a** **c** ) ( **b** **d** ) ( **e** )

## Mintapélda megoldása lépcsős táblán (4)

*előzetes szimbolikus állapottábla*

| aktuális állapot | következő állapot/kimenet |            |
|------------------|---------------------------|------------|
|                  | $X=0$                     | $X=1$      |
| <b>a</b>         | <b>c/0</b>                | <b>b/1</b> |
| <b>b</b>         | <b>d/1</b>                | <b>e/0</b> |
| <b>c</b>         | <b>a/0</b>                | <b>d/1</b> |
| <b>d</b>         | <b>b/1</b>                | <b>e/0</b> |
| <b>e</b>         | <b>e/0</b>                | <b>a/1</b> |

**(ac)  $\rightarrow s_1$**

**(bd)  $\rightarrow s_2$**

**e  $\rightarrow s_3$**



*összevont szimbolikus állapottábla*

| aktuális állapot | következő állapot/kimenet |         |
|------------------|---------------------------|---------|
|                  | $X=0$                     | $X=1$   |
| $s_1$            | $s_1/0$                   | $s_2/1$ |
| $s_2$            | $s_2/1$                   | $s_3/0$ |
| $s_3$            | $s_3/0$                   | $s_1/1$ |

## Állapot összevonás ekvivalencia-típusú reláció alkalmazásával: összefoglaló

### Lépések:

- 1. az ekvivalens és antivalens állapot párok megkeresése lépcsős tábla segítségével,*
- 2. a maximális ekvivalencia osztályok meghatározása,*
- 3. a maximális ekvivalencia osztályoknak megfelelő állapotokkal az összevont állapottábla elkészítése.*

## Állapot összevonás nem teljesen specifikált előzetes szimbolikus állapottáblán

*A nem megkülönböztethetőség, mint feltétel értelmezése a nem teljesen specifikált, előzetes szimbolikus állapottáblán:*

a nem teljesen specifikált előzetes, szimbolikus állapottáblán két állapot nem megkülönböztethető, ha bemeneti kombinációként megegyeznek a kimeneti kombinációk (amennyiben mindkettőre specifikálva vannak), és a következő állapotok is nem megkülönböztethetők (ha mindkettőre specifikálva vannak).

## Állapot összevonás nem teljesen specifikált előzetes szimbolikus állapottáblán: példa1.

**Példa1.:** állapot összevonás egy korábbi, nem teljesen specifikált előzetes szimbolikus állapottáblán

Egy lehetséges megoldás:

tegyük teljesen specifikálttá, és csináljunk összevonást ekvivalencia relációkkal!

| $X1 \backslash X2$<br>bem.<br>akt. áll. | szimb.köv.áll. / kim. |       |       |       |
|-----------------------------------------|-----------------------|-------|-------|-------|
|                                         | 0 0                   | 0 1   | 1 0   | 1 1   |
| a                                       | (a)/0                 | b/0   | c/0   | -/-   |
| b                                       | a/0                   | (b)/0 | -/-   | d/0   |
| c                                       | a/0                   | -/-   | (c)/0 | e/1   |
| d                                       | -/-                   | b/0   | c/0   | (d)/0 |
| e                                       | -/-                   | b/0   | c/0   | (e)/1 |

| $X1 \backslash X2$<br>bem.<br>akt. áll. | szimb.köv.áll. / kim. |                     |                     |                     |
|-----------------------------------------|-----------------------|---------------------|---------------------|---------------------|
|                                         | 0 0                   | 0 1                 | 1 0                 | 1 1                 |
| <i>a</i>                                | $\textcircled{a}/0$   | <i>b</i> /0         | <i>c</i> /0         | -/-                 |
| <i>b</i>                                | <i>a</i> /0           | $\textcircled{b}/0$ | -/-                 | <i>d</i> /0         |
| <i>c</i>                                | <i>a</i> /0           | -/-                 | $\textcircled{c}/0$ | <i>e</i> /1         |
| <i>d</i>                                | -/-                   | <i>b</i> /0         | <i>c</i> /0         | $\textcircled{d}/0$ |
| <i>e</i>                                | -/-                   | <i>b</i> /0         | <i>c</i> /0         | $\textcircled{e}/1$ |

→ (*abd*), (*ce*)



| $X1 \backslash X2$<br>bem.<br>akt. áll. | szimb.köv.áll. / kim. |                     |                     |                     |
|-----------------------------------------|-----------------------|---------------------|---------------------|---------------------|
|                                         | 0 0                   | 0 1                 | 1 0                 | 1 1                 |
| <i>a</i>                                | $\textcircled{a}/0$   | <i>b</i> /0         | <i>c</i> /0         | <b>d</b> /0         |
| <i>b</i>                                | <i>a</i> /0           | $\textcircled{b}/0$ | <b>c</b> /0         | <i>d</i> /0         |
| <i>c</i>                                | <i>a</i> /0           | <b>b</b> /0         | $\textcircled{c}/0$ | <i>e</i> /1         |
| <i>d</i>                                | <b>a</b> /0           | <i>b</i> /0         | <i>c</i> /0         | $\textcircled{d}/0$ |
| <i>e</i>                                | <b>a</b> /0           | <i>b</i> /0         | <i>c</i> /0         | $\textcircled{e}/1$ |

## Az állapot-kompatibilitás

Egy nem teljesen specifikált szimbolikus előzetes állapotáblával megadott hálózat adott állapotához tartozó specifikációs bemeneti sorozat az, amelyre a hálózat minden állapotátmenete és kimenete specifikálva van.

Két szimbolikus állapot a nem teljesen specifikált hálózat állapotábláján csak akkor megkülönböztethető, ha létezik *legalább* egy olyan specifikált bemeneti sorozat, amely mindkét állapotra *érvényes*, és amelynek *legalább* egy *elemére más kimeneti kombináció adódik*. Ha ilyen specifikációs bemeneti sorozat nem létezik, a két állapot nem megkülönböztethető.

Továbbá ha a kiválasztott két állapotra létezik olyan bemeneti kombináció, amelyre *vagy a kimenetek, vagy a következő állapotok, vagy mindkettő* specifikálva vannak, a két állapot akkor nem megkülönböztethető, ha a specifikált kimeneti kombinációk bemeneti kombinációnként megegyeznek, a specifikált következő állapotok pedig nem megkülönböztethetők.

## A „nem megkülönböztethetőség”(NMK), mint reláció

### Tulajdonságok:

1. *reflexív*
2. *szimmetrikus*
3. *nem tranzitív*

Az ilyen relációkat ***kompatibilitás-típusú relációknak*** nevezzük. A nem teljesen specifikált hálózat állapotpárjaira fennálló 'nem megkülönböztethetőség' tulajdonságot röviden *kompatibilitásnak*, illetve a pár tagjait *kompatibilis állapotoknak* fogjuk nevezni.

## Jelölések a lépcsős táblán

- $a \sim b$  :  $a'$  és  $b'$  állapotok kompatibilisek
- $a \not\sim b$  :  $a'$  és  $b'$  állapotok nem kompatibilisek
- *feltételes kompatibilitás*:  $ab, cd \dots$  az a két állapot, melyekre ez a bejegyzés vonatkozik, feltételesen kompatibilis, azaz csak akkor kompatibilis, ha  $a \sim b$  és  $c \sim d \dots$

## A kompatibilitás elégséges feltételei

*A kompatibilitás elégséges feltételei a következők:*

- ha nincs olyan bemeneti kombináció, amelyre mindkét állapotból specifikált következő állapot és specifikált kimenet lenne az állapottáblán, akkor a két állapot kompatibilis,*
- ha pedig létezik mindkét állapotra specifikált kimeneti kombinációt és következő állapotot definiáló bemeneti kombináció, és erre a két állapothoz tartozó kimeneti kombinációk megegyeznek, valamint a két állapothoz tartozó következő állapotok kompatibilisek, akkor a két állapot kompatibilis.*

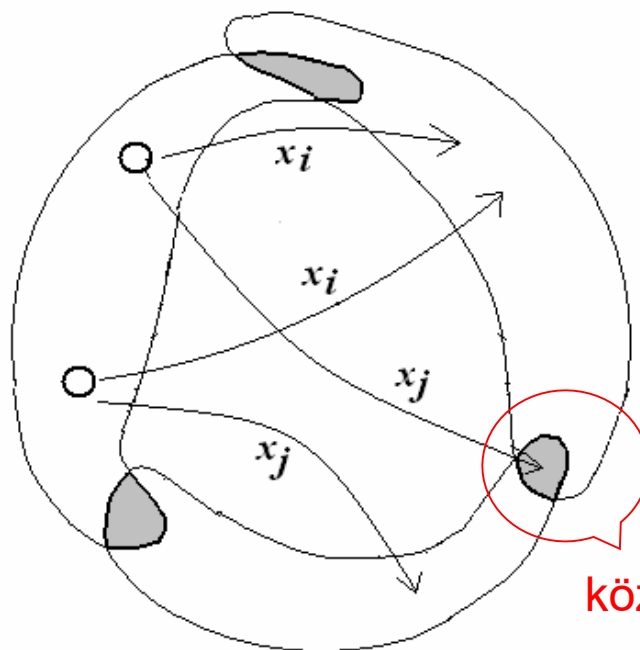
## A kompatibilitási osztályok zárt halmaza

A kompatibilitási reláció az állapothalmazt nem diszjunkt osztályokra bontja, azaz lehetnek az osztályoknak közös elemeik is. Egy ilyen osztály valamennyi lehetséges állapotpárjára fennáll a kompatibilitás. Ezek az osztályok akkor maximálisak, ha további elemek egyetlen osztályba sem vonhatók be.

A kompatibilitási osztályok egy adott halmaza zárt, ha a halmazban szereplő bármelyik osztály tetszőleges két állapotából kiindulva minden olyan bemeneti kombinációra, amely mindkét állapotból specifikált következő állapotot ír elő, a következő állapotok is együtt szerepelnek a halmaznak *legalább egy osztályában*.

*A kompatibilitási reláció jellegzetessége:  
a nem diszjunkt részhalmazok zártsága*

**Példa:**



közös osztályrész!

## **Kevesebb, vagy kisebb állapot-számú osztályból álló zárt kompatibilitási osztályhalmaz keresése**

A maximális kompatibilitási osztályok megtalálása után a nem teljes specifikáció lehetőséget teremt arra, hogy az állapotok teljes lefedése és a zártság megőrzése mellett egyszerűbb kompatibilitási halmazrendszert válasszunk.

Ez azt jelenti, hogy úgy döntünk a közömbös bejegyzésekről, hogy döntésünk

- vagy kevesebb kompatibilitási osztályból álló,
- vagy az egyes osztályokban kevesebb állapotból álló osztályhalmazt eredményezzen.

## Kevesebb, vagy kisebb állapot-számú osztályból álló zárt kompatibilitási osztályhalmaz keresése

*Követelmények:*

- a maximális kompatibilitási osztályoknak továbbra is zárt halmazrendszert kell alkotni
- minden állapotnak szerepelnie kell legalább egy osztályban

## Kevesebb, vagy kisebb állapot-számú osztályból álló zárt kompatibilitási osztályhalmaz keresése

Vizsgálat: van-e olyan kompatibilitási osztály, amelynek valamennyi állapota szerepel valamely más osztályban is?

- Ha így van, megkísérelhetjük elhagyni ezt az osztályt. Ez akkor lehetséges, ha az osztály elhagyása után is zárt marad a kompatibilitási osztályok halmaza.
- Ha a zártság nem tartható fenn, akkor visszatesszük az elhagyni kívánt osztályt, és a többszörösen szereplő állapotok egyes osztályokból való elhagyásával próbálkozunk.

Ha találunk a teljes lefedettség és a zártság fenntartásával elhagyható állapotokat, akkor egyszerűbb összevont állapottáblát kapunk.

A fentiekből következik, hogy több megoldás is kínálkozhat, ezek közül kell választanunk a megvalósítandó összevont állapottáblát.

## Állapot összevonás nem teljesen specifikált előzetes szimbolikus állapottáblán: példa2.

**Példa2.:** alkalmazzuk a kompatibilitás-típusú relációt az előző példában (példa1) bemutatott állapottáblán:

| aktuális<br>állapot | következő<br>állapot/kimenet |            |            |            |
|---------------------|------------------------------|------------|------------|------------|
|                     | $X_1 X_2$                    |            |            |            |
|                     | <i>00</i>                    | <i>01</i>  | <i>10</i>  | <i>11</i>  |
| <i>a</i>            | <i>a/0</i>                   | <i>b/0</i> | <i>c/0</i> | <i>-/-</i> |
| <i>b</i>            | <i>a/0</i>                   | <i>b/0</i> | <i>-/-</i> | <i>d/0</i> |
| <i>c</i>            | <i>a/0</i>                   | <i>-/-</i> | <i>c/0</i> | <i>e/1</i> |
| <i>d</i>            | <i>-/-</i>                   | <i>b/0</i> | <i>c/0</i> | <i>d/0</i> |
| <i>e</i>            | <i>-/-</i>                   | <i>b/0</i> | <i>c/0</i> | <i>e/1</i> |

## Állapot összevonás nem teljesen specifikált előzetes szimbolikus állapottáblán: példa2.

*előzetes szimbolikus állapottábla*

| aktuális állapot | következő állapot/kimenet |             |             |             |
|------------------|---------------------------|-------------|-------------|-------------|
|                  | $X_1 X_2$                 |             |             |             |
|                  | 00                        | 01          | 10          | 11          |
| <i>a</i>         | <i>a</i> /0               | <i>b</i> /0 | <i>c</i> /0 | -/-         |
| <i>b</i>         | <i>a</i> /0               | <i>b</i> /0 | -/-         | <i>d</i> /0 |
| <i>c</i>         | <i>a</i> /0               | -/-         | <i>c</i> /0 | <i>e</i> /1 |
| <i>d</i>         | -/-                       | <i>b</i> /0 | <i>c</i> /0 | <i>d</i> /0 |
| <i>e</i>         | -/-                       | <i>b</i> /0 | <i>c</i> /0 | <i>e</i> /1 |



|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| <b>b</b> | ~        |          |          |          |
| <b>c</b> | ~        | /~       |          |          |
| <b>d</b> | ~        | ~        | /~       |          |
| <b>e</b> | ~        | /~       | ~        | /~       |
|          | <b>a</b> | <b>b</b> | <b>c</b> | <b>d</b> |

*lépcsős tábla*

## Állapot összevonás nem teljesen specifikált előzetes szimbolikus állapottáblán: példa2.

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| <b>b</b> | $\sim$   |          |          |          |
| <b>c</b> | $\sim$   | $/\sim$  |          |          |
| <b>d</b> | $\sim$   | $\sim$   | $/\sim$  |          |
| <b>e</b> | $\sim$   | $/\sim$  | $\sim$   | $/\sim$  |
|          | <b>a</b> | <b>b</b> | <b>c</b> | <b>d</b> |

kompatibilis párok:

(a b), (a c), (a d), (a e), (b d), (c,e)

maximális kompatibilitási osztályok:

(a b d), (a c e)

## Állapot összevonás nem teljesen specifikált előzetes szimbolikus állapottáblán: példa2.

### Konklúzió

két redukált, zárt osztályhalmaz:

- 1) **(a b d) (c e),**
- 2) **(a c e) (b d)**

Az első osztályhalmaz zártságáról az állapottábla alapján meggyőződhetünk, és beláthatjuk, hogy az **(a b d)** minden eleme bemeneti kombinációnként ugyanabba az osztályba képződik le, illetve ez a **(c e)** osztály elemeire is igaz. Hasonlóan bizonyítható a második osztályhalmaz zártsága is. Ebből az következik, hogy a példának kétféle állapot összevonása is jó megoldáshoz vezet.

## Állapot összevonás nem teljesen specifikált előzetes szimbolikus állapottáblán: példa2.

Megoldás(1):

- (a b d) (c e),



(a b d)  $\rightarrow S_1$

(c e)  $\rightarrow S_2$

| aktuális<br>állapot | következő állapot/kimenet |         |         |         |
|---------------------|---------------------------|---------|---------|---------|
|                     | $X_1X_2$                  |         |         |         |
|                     | 00                        | 01      | 10      | 11      |
| $S_1$               | $S_1/0$                   | $S_1/0$ | $S_2/0$ | $S_1/0$ |
| $S_2$               | $S_1/0$                   | $S_1/0$ | $S_2/0$ | $S_2/1$ |

Megoldás(2):

- (a c e) (b d)



(a c e)  $\rightarrow S_1$

(b d)  $\rightarrow S_2$

| aktuális<br>állapot | következő állapot/kimenet |         |         |         |
|---------------------|---------------------------|---------|---------|---------|
|                     | $X_1X_2$                  |         |         |         |
|                     | 00                        | 01      | 10      | 11      |
| $S_1/0$             | $S_1/0$                   | $S_2/0$ | $S_1/0$ | $S_1/1$ |
| $S_2/0$             | $S_1/0$                   | $S_2/0$ | $S_1/0$ | $S_2/0$ |

## Állapotkódolási módszerek

### ➤ Szinkron sorrendi hálózatok:

- nincs versenyhelyzet veszély, így az állapotkódolás arra irányul, hogy a legegyszerűbb struktúrát alakítsuk ki.

### ➤ Aszinkron sorrendi hálózatok:

- *legfontosabb cél a kritikus versenyhelyzetek elkerülése*

## Állapotkódolási módszerek

### ➤ Szinkron sorrendi hálózatok állapotkódolási módszerei

- az ,1'-es súlyú állapotkódolás elve (ism.)
- a szomszédos kódolás elve
- a helyettesítési tulajdonságú partíció elve alapján végzett kódolás

### ➤ Aszinkron sorrendi hálózatok állapotkódolási módszerei

- instabil állapotok beillesztésének elve
- a Tracey-Unger módszer elve

## Szinkron sorrendi hálózatok állapotkódolási módszerei (1)

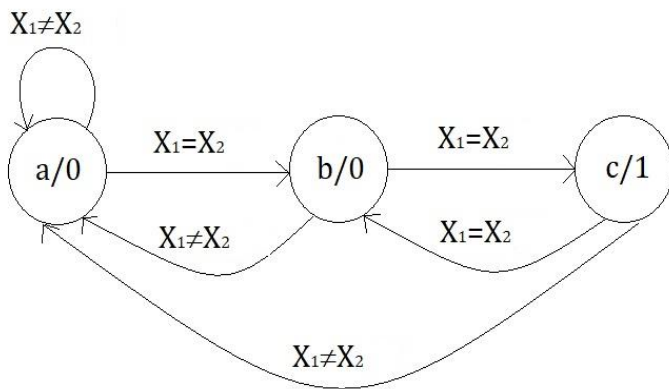
➤ az ,1'-es súlyú állapotkódolás („bit per state”) elve (ism.)

(...korábban...)

Az olyan kódolást, melyben a kódszóban csak egyetlen bit helyén szerepel '1'-es érték (a többi '0'), ,1'-es súlyú kódolásnak nevezzük

## Szinkron sorrendi hálózatok állapotkódolási módszerei (1)

Egy korábbi példa ,1'-es súlyozású kódolás alkalmazására:



|       | $Q_a$ | $Q_b$ | $Q_c$ |
|-------|-------|-------|-------|
| $a :$ | 1     | 0     | 0     |
| $b :$ | 0     | 1     | 0     |
| $c :$ | 0     | 0     | 1     |



$$D_a = R + \bar{R} (Q_a \bar{E} + Q_b \bar{E} + Q_c \bar{E}) = R + \bar{E} (Q_a + Q_b + Q_c) \\ = R + \bar{E}$$

$$D_b = \bar{R} E (Q_a + Q_c)$$

$$D_c = \bar{R} Q_b E$$

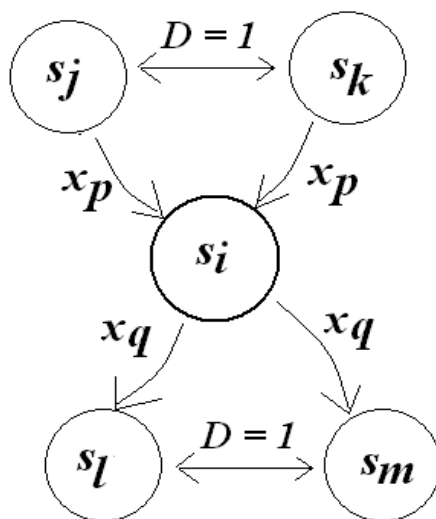
## Szinkron sorrendi hálózatok állapotkódolási módszerei (2)

### ➤ *a szomszédos kódolás elve*

Egyszerűbb lesz a hálózat, ha egy adott állapot következő állapotainak kódjai bemenő kombinációként szomszédosak, illetve csak egy szekunder változóban különböznek egymástól, azaz a kódok közötti távolság egységnyi ( $D=1$ ). Ugyancsak előnyös, ha azok az állapotok, amelyek valamely adott állapotnak az elődjei, bemenő kombinációként szomszédos kódúak.

## Szinkron sorrendi hálózatok állapotkódolási módszerei (2)

Példa:



- szomszédos kódolást kifejező állapotgráf a leírt előnyös kódolási helyzetekkel

| aktuális állapot | következő állapot |       |
|------------------|-------------------|-------|
|                  | $X=0$             | $X=1$ |
| $a$              | $c$               | $d$   |
| $b$              | $a$               | $e$   |
| $c$              | $a$               | $b$   |
| $d$              | $c$               | $e$   |
| $e$              | $a$               | $b$   |

- előzetes szimbolikus állapottábla

## Szinkron sorrendi hálózatok állapotkódolási módszerei (2)

| állapotok | állapotok: „utódok” |       |
|-----------|---------------------|-------|
|           | $X=0$               | $X=1$ |
| $a$       | $c$                 | $d$   |
| $b$       | $a$                 | $e$   |
| $c$       | $a$                 | $b$   |
| $d$       | $c$                 | $e$   |
| $e$       | $a$                 | $b$   |

| állapotok | állapotok: „elődök” |       |
|-----------|---------------------|-------|
|           | $X=0$               | $X=1$ |
| $a$       | $b,c,e$             |       |
| $b$       |                     | $c,e$ |
| $c$       | $a,d$               |       |
| $d$       | $a$                 |       |
| $e$       |                     | $b,d$ |

- szomszédos kódolás elve alapján átdolgozott „utódok” és „elődök” táblázatok

## Szinkron sorrendi hálózatok állapotkódolási módszerei (2)

- az állapotok elhelyezése Karnaugh táblán, az előnyös szomszédosságok legnagyobb arányú biztosítására

Közös utódja van  $X = 0$ -ra:

b,c  
b,e  
c,e  
a,d

Közös utódja van  $X = 1$ -re:

b,d  
c,e

Közös elődje van  $X = 0$ -ra:

c,d

Közös elődje van  $X = 1$ -re

-

Szomszédos kódok:

| <i>mindkettő</i> | <i>közös utód</i> | <i>közös előd</i> |
|------------------|-------------------|-------------------|
| —                | a,d               | c,d               |
|                  | b,c               |                   |
|                  | b,d               |                   |
|                  | b,e               |                   |
|                  | c,e               |                   |

|    |          |   |   |   |   |
|----|----------|---|---|---|---|
|    | Q1<br>Q2 | 0 | 0 | 1 | 1 |
|    |          | 0 | 1 | 1 | 0 |
| Q3 | 0        | e | b |   |   |
|    | 1        | c | d | a |   |

## Szinkron sorrendi hálózatok állapotkódolási módszerei

(2)

- egy lehetséges kód kiosztás

|      | $Q_1$ | $Q_2$ | $Q_3$ |
|------|-------|-------|-------|
| $a:$ | 1     | 1     | 1     |
| $b:$ | 0     | 1     | 0     |
| $c:$ | 0     | 0     | 1     |
| $d:$ | 0     | 1     | 1     |
| $e:$ | 0     | 0     | 0     |
| -    | 1     | 0     | 0     |
| -    | 1     | 1     | 0     |
| -    | 1     | 0     | 1     |



| kódolt aktuális<br>állapot,<br>$Q_1^n \quad Q_2^n \quad Q_3^n$ | kódolt következő állapot |       |       |       |       |       |
|----------------------------------------------------------------|--------------------------|-------|-------|-------|-------|-------|
|                                                                | X = 0                    |       |       | X = 1 |       |       |
|                                                                | $D_1$                    | $D_2$ | $D_3$ | $D_1$ | $D_2$ | $D_3$ |
| 1 1 1                                                          | 0                        | 0     | 1     | 0     | 1     | 1     |
| 0 1 0                                                          | 1                        | 1     | 1     | 0     | 0     | 0     |
| 0 0 1                                                          | 1                        | 1     | 1     | 0     | 1     | 0     |
| 0 1 1                                                          | 0                        | 0     | 1     | 0     | 0     | 0     |
| 0 0 0                                                          | 0                        | 1     | 0     | 1     | 1     | 1     |
| 1 0 0                                                          | -                        | -     | -     | -     | -     | -     |
| 1 1 0                                                          | -                        | -     | -     | -     | -     | -     |
| 1 0 1                                                          | -                        | -     | -     | -     | -     | -     |

- kódolt állapottábla

(...)

## Szinkron sorrendi hálózatok állapotkódolási módszerei (3)

➤ *A helyettesítési tulajdonságú partíció elve alapján végzett kódolás*

Alap: az állapothalmazon értelmezett partíciópárok elmélete

Ha egy szinkron sorrendi hálózat állapotváltozóit *csoportokra* bontjuk, akkor a csoportokhoz *komponenseket* rendelhetünk. Minden egyes komponenshez *két partíciót* tartozik:

- az egyik partíció azon állapotokat sorolja egy osztályba, amelyeket a komponens egyformán kódol,
- a másik partíció azokat az állapotokat sorolja egy osztályba, amelyeket a komponensre kapcsolódó környezet egyformán kódol.

## Szinkron sorrendi hálózatok állapotkódolási módszerei (3)

A környezeti partíció és a komponens partíció úgynevezett *partíciópárt* alkot.

Mivel a komponens a következő állapotának kialakításakor egy adott bemeneti kombináció esetén nem tesz különbséget két, a környezete által azonosan kódolt állapot között, a környezeti partíció ugyanazon osztályába tartozó állapotok következő állapotai bemeneti kombinációként a komponens partíciónak is ugyanazon blokkjában vannak.

Ezért nevezzük ezt a partíció kettőst *partíciópárnak*. A komponens partíciók metszete a legfinomabb partíciót eredményezi.

Ha egy komponens bemeneteire nem kapcsolódnak más komponensek állapotváltozói, akkor a komponens partíció önmagával alkot partíciópárt.

Az ilyen partíciót helyettesítési tulajdonságú (HT) partíciónak nevezik.

(...)

## Aszinkron sorrendi hálózatok állapotkódolási módszerei (1)

### ➤ *instabil állapotok beillesztésének elve*

A módszer lényege: instabil állapotokkal bővítjük az összevont állapottáblát.

Az instabil állapotokkal való bővítést úgy kell megoldani, hogy a stabil állapotok egymás utáni sorrendje ne változzék, ugyanakkor az új instabil állapotokat úgy kell kódolni, hogy az egymás után következő tranziens kódok között csak egy szekunder változó értékében legyen különbség.

## Aszinkron sorrendi hálózatok állapotkódolási módszerei (1)

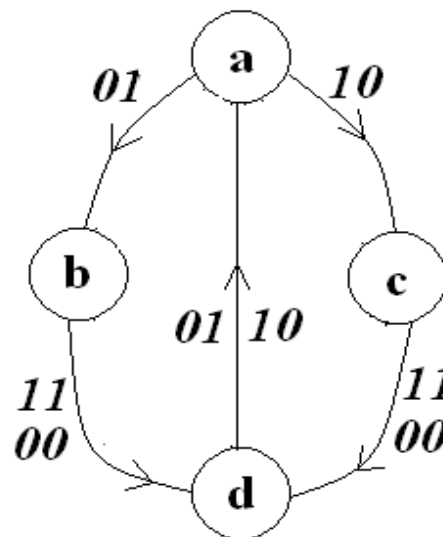
Példa:

|   | 00 | 01 | 10 | 11 |
|---|----|----|----|----|
| a | a  | b  | c  | -  |
| b | d  | b  | -  | d  |
| c | d  | -  | c  | d  |
| d | d  | a  | a  | d  |

áll.kódok

|   | 0 | 1 |
|---|---|---|
| 0 | a | b |
| 1 | c | d |

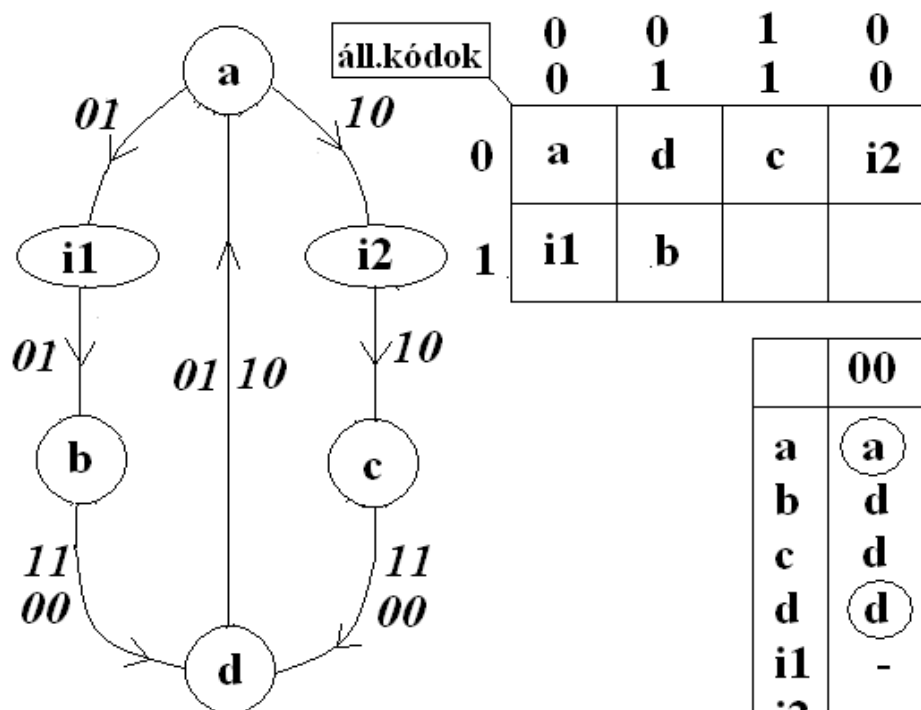
az a-val nem szomszédos



## Aszinkron sorrendi hálózatok állapotkódolási módszerei (1)

Megoldás:

- i1 és i2 beillesztése



## Aszinkron sorrendi hálózatok állapotkódolási módszerei (2)

### ➤ *a Tracey-Unger módszer elve*

Aszinkron sorrendi hálózatokban a szekunder változók között kritikus versenyhelyzet akkor áll elő, ha egy stabil állapotból kiindulva megváltoztatjuk a bemeneti kombinációt, és ennek hatására olyan átmeneti állapotkód áll elő, amelynek sorában és az adott bemeneti kombináció oszlopában ez az állapotkód szerepel. A nem kívánt átmeneti állapotkódot *hazárd kódnak* nevezzük.

## Aszinkron sorrendi hálózatok állapotkódolási módszerei (2)

*A Tracey-Unger módszer lényege, hogy a normális (tervezett) állapotátmenethez tartozó kiinduló és cél állapotok kódjai legalább egy adott szekunder változóban megegyezzenek, és ebben a változóban mindketten különbözzenek a hazard kódtól.*

(Ilyenkor ugyanis ez a szekunder változó az átmenet során állandó marad, és így soha sem áll elő a hazard állapot kódja.)

## Aszinkron sorrendi hálózatok állapotkódolási módszerei (2)

Példa:

| akt.áll | köv.áll / z |        |        |        |
|---------|-------------|--------|--------|--------|
|         | D C         |        |        |        |
|         | 0 0         | 0 1    | 1 0    | 1 1    |
| s1      | s1 / 0      | s1 / 0 | s2 / 0 | s1 / 0 |
| s2      | s1 / 0      | - / -  | s2 / 0 | s3 / 1 |
| s3      | s4 / 1      | s3 / 1 | s3 / 1 | s3 / 1 |
| s4      | s4 / 1      | s1 / 0 | s3 / 1 | - / -  |

A megváltozott bemeneti kombináció oszlopában találjuk a tervezett új stabil állapot szimbólumát, valamint a stabilizálódott szimbólumot is. Az oszlopban szereplő minden más stabil állapot „leselkedő” *potenciális hazard*.

*Például : ha (00, s1) állapotból az (10, s2) állapotba mennénk, az s3 leselkedő, azaz el kéne kerülni, hogy a kódja tranziensként megjelenjen.*

## Aszinkron sorrendi hálózatok állapotkódolási módszerei (2)

Megoldás:

| akt.áll | köv.áll / z |        |        |        |
|---------|-------------|--------|--------|--------|
|         | D C<br>0 0  | 0 1    | 1 0    | 1 1    |
| s1      | Ⓢ1 / 0      | Ⓢ1 / 0 | s2 / 0 | Ⓢ1 / 0 |
| s2      | s1 / 0      | - / -  | Ⓢ2 / 0 | s3 / 1 |
| s3      | s4 / 1      | Ⓢ3 / 1 | Ⓢ3 / 1 | Ⓢ3 / 1 |
| s4      | Ⓢ4 / 1      | s1 / 0 | s3 / 1 | - / -  |

A lehetséges bemeneti változások szerint listába vesszük a tervezett stabil-stabil átmeneteket, a leselkedő feltüntetésével.

| 00->01            | 00->10            | 01->00            | 01->11 | 10->00                                | 10->11            | 11->01 | 11->10                                |
|-------------------|-------------------|-------------------|--------|---------------------------------------|-------------------|--------|---------------------------------------|
| s4->s1<br>s3<br>1 | s1->s2<br>s3<br>2 | s3->s4<br>s1<br>4 | -      | s2->s1<br>s4<br>5                     | s2->s3<br>s1<br>6 | -      | <del>s1-&gt;s2</del><br><del>s3</del> |
|                   | s4->s3<br>s2<br>3 |                   |        | <del>s3-&gt;s4</del><br><del>s1</del> |                   |        |                                       |
|                   |                   |                   |        |                                       |                   |        |                                       |

## Aszinkron sorrendi hálózatok állapotkódolási módszerei (2)

| 00->01            | 00->10            | 01->00            | 01->11 | 10->00                                | 10->11            | 11->01 | 11->10                                |
|-------------------|-------------------|-------------------|--------|---------------------------------------|-------------------|--------|---------------------------------------|
| s4->s1<br>s3<br>1 | s1->s2<br>s3<br>2 | s3->s4<br>s1<br>4 | -      | s2->s1<br>s4<br>5                     | s2->s3<br>s1<br>6 | -      | <del>s1-&gt;s2</del><br><del>s3</del> |
|                   | s4->s3<br>s2<br>3 |                   |        | <del>s3-&gt;s4</del><br><del>s1</del> |                   |        |                                       |
|                   |                   |                   |        |                                       |                   |        |                                       |

- ahány különböző szabály, annyi szekunder változót írunk fel, és annak az állapotait a szabály alapján határozzuk meg

←

|    | Y1<br>1 2 | Y2<br>1 2 | Y3<br>1 2 | Y4<br>1 2 | Y5<br>1 2 | Y6<br>1 2 |
|----|-----------|-----------|-----------|-----------|-----------|-----------|
| s1 | 0 1       | 0 1       | - -       | 1 0       | 0 1       | 1 0       |
| s2 | - -       | 0 1       | 1 0       | - -       | 0 1       | 0 1       |
| s3 | 1 0       | 1 0       | 0 1       | 0 1       | - -       | 0 1       |
| s4 | 0 1       | - -       | 0 1       | 0 1       | 1 0       | - -       |

- minden szekunder változóra mindkét lehetséges választást felírjuk

←

## Aszinkron sorrendi hálózatok állapotkódolási módszerei (2)

|    | Y1<br>1 2 | Y2<br>1 2 | Y3<br>1 2 | Y4<br>1 2 | Y5<br>1 2 | Y6<br>1 2 |
|----|-----------|-----------|-----------|-----------|-----------|-----------|
| s1 | 0 1       | 0 1       | - -       | 1 0       | 0 1       | 1 0       |
| s2 | - -       | 0 1       | 1 0       | - -       | 0 1       | 0 1       |
| s3 | 1 0       | 1 0       | 0 1       | 0 1       | - -       | 0 1       |
| s4 | 0 1       | - -       | 0 1       | 0 1       | 1 0       | - -       |

- összevonható változók

- két szekunder változó csoport

|    | $Y_p$ | $Y_z$ |
|----|-------|-------|
| s1 | 0     | 0     |
| s2 | 0     | 1     |
| s3 | 1     | 1     |
| s4 | 1     | 0     |

## Aszinkron sorrendi hálózatok állapotkódolási módszerei (2)

|             |  |  |  |  |      |       |       |
|-------------|--|--|--|--|------|-------|-------|
|             |  |  |  |  |      | $Y_p$ | $Y_z$ |
|             |  |  |  |  | $s1$ | 0     | 0     |
|             |  |  |  |  | $s2$ | 0     | 1     |
|             |  |  |  |  | $s3$ | 1     | 1     |
| köv.áll / z |  |  |  |  | $s4$ | 1     | 0     |

| akt.áll | D C    |        |        |        | ↓ | $Y_p^v$ $Y_z^v$ | $Y_p Y_z / Z$ |         |        |         |
|---------|--------|--------|--------|--------|---|-----------------|---------------|---------|--------|---------|
|         | 0 0    | 0 1    | 1 0    | 1 1    |   |                 | 0 0           | 0 1     | 1 0    | 1 1     |
| s1      | s1 / 0 | s1 / 0 | s2 / 0 | s1 / 0 | ⇒ | 0 0             | 00 / 0        | 00 / 0  | 01 / 0 | 00 / 0  |
| s2      | s1 / 0 | - / -  | s2 / 0 | s3 / 1 |   | 0 1             | 00 / 0        | - - / - | 01 / 0 | 11 / 1  |
| s3      | s4 / 1 | s3 / 1 | s3 / 1 | s3 / 1 |   | 1 1             | 10 / 1        | 11 / 1  | 11 / 1 | 11 / 1  |
| s4      | s4 / 1 | s1 / 0 | s3 / 1 | - / -  |   | 1 0             | 10 / 1        | 00 / 0  | 11 / 1 | - - / - |

- előzetes szimbolikus állapotábra

- a kódolt állapotábra

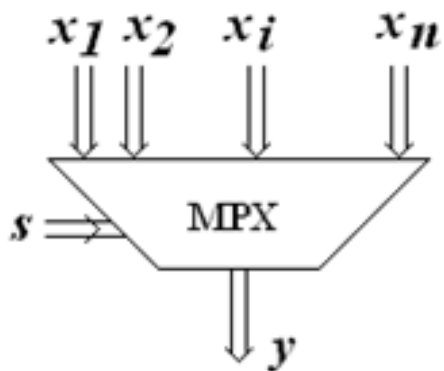
(...)

## Az összetett digitális egységek csoportjai

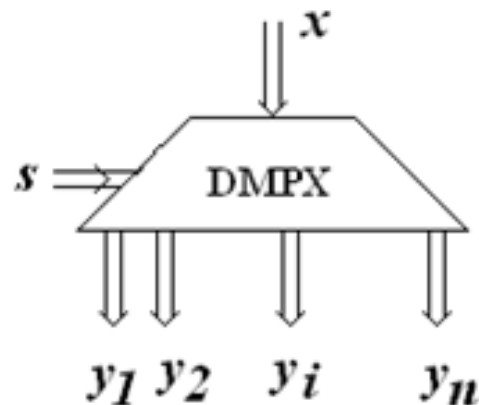
- Multiplexerek, demultiplexerek:
  - *feladatuk adatút szakaszok kijelölése*
  
- Regiszterek:
  - *adatokat tárolnak, és ezek elérését biztosítják*
  
- Funkciós egységek:
  - *adatok közötti műveleteket végeznek*

## Multiplexerek, demultiplexerek

- feladatuk adatút szakaszok kijelölése*



Multiplexer általános sémája



Demultiplexer általános sémája

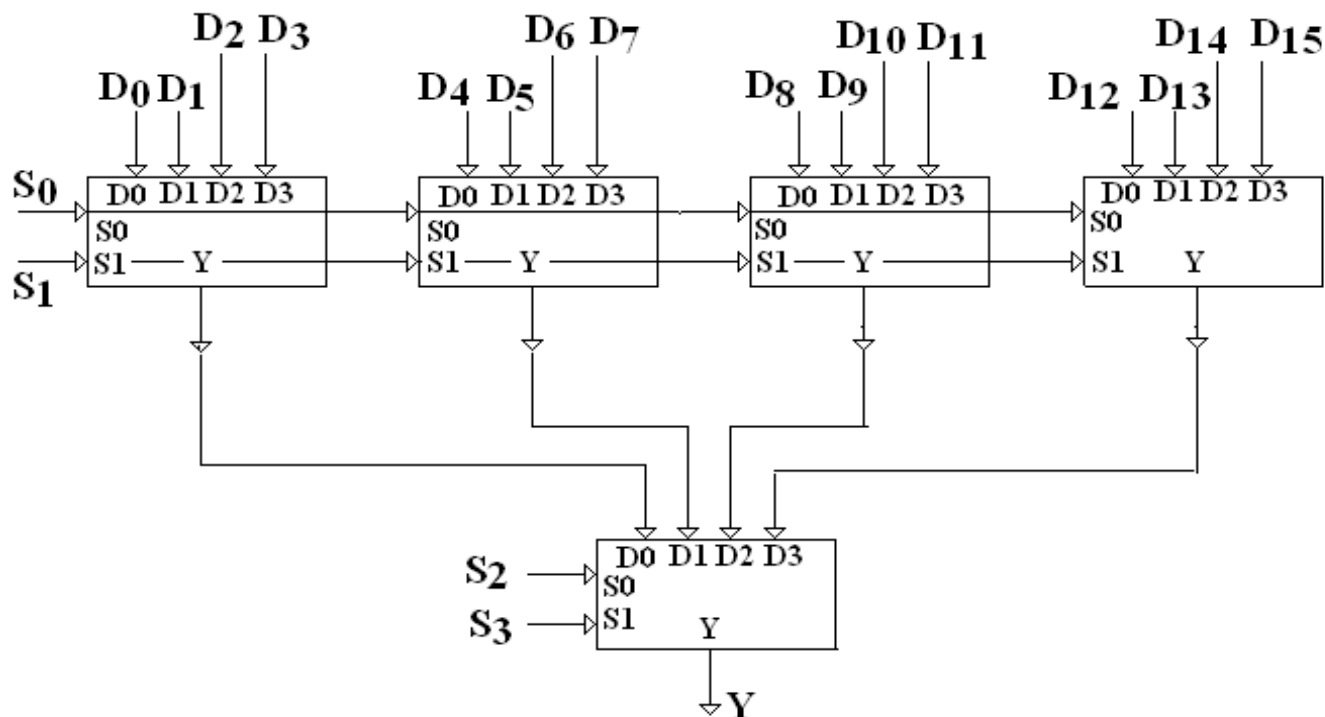
## Multiplexerek

- *A multiplexerek, mint programozható logikai hálózatok (ism.):*
  - a mintermes kanonikus alakban megadott függvény mintermjait a címző (vezérlő) bemenetekre adott címek képviselik, és a megcímzett adatbemenetre rá kell kapcsolnunk az adott mintermhez tartozó logikai értéket. Ezeknek a logikai konstansoknak a bemenetekre való kapcsolását tekinthetjük a multiplexerek programozásának.

(...)

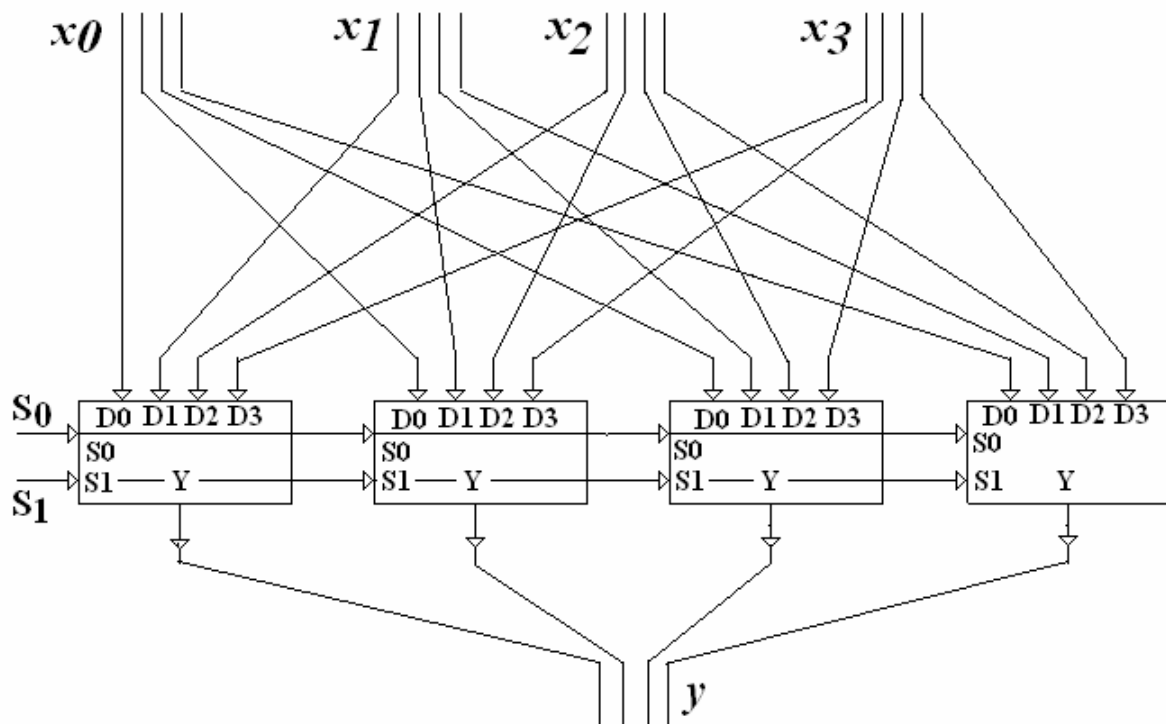
## Multiplexerek

- Alkalmazási példa1.: bővítés a bemenetek számának növelésére (MPX4-1 multiplexerek felhasználásával)*



## Multiplexerek

- Alkalmazási példa2.: bővítés sínek közötti választás céljából(MPX4-1 alapegységekből)*

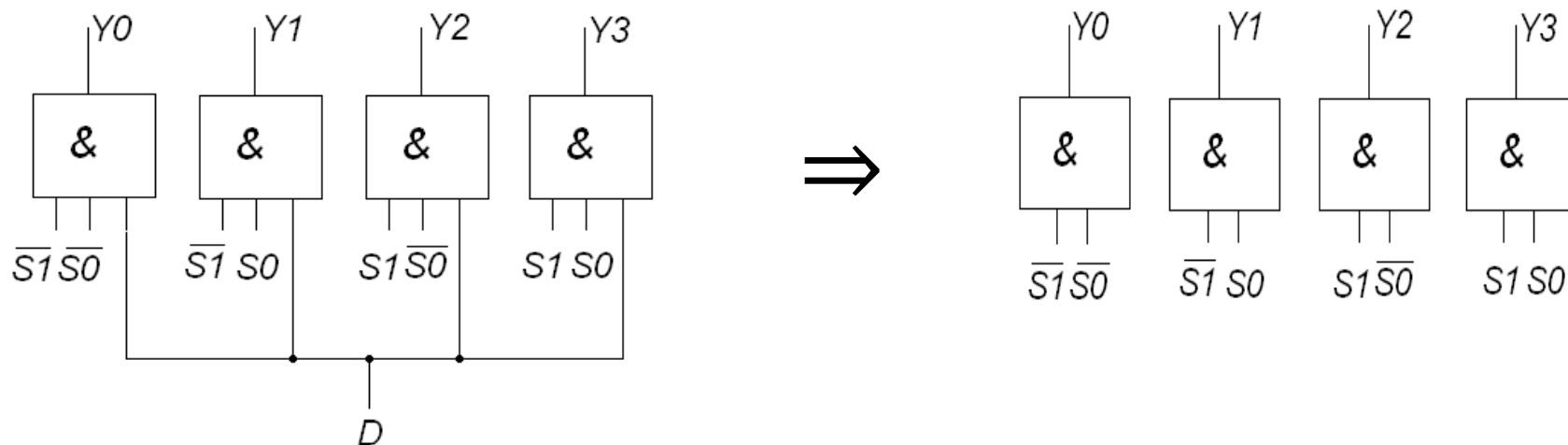


## Demultiplexerek

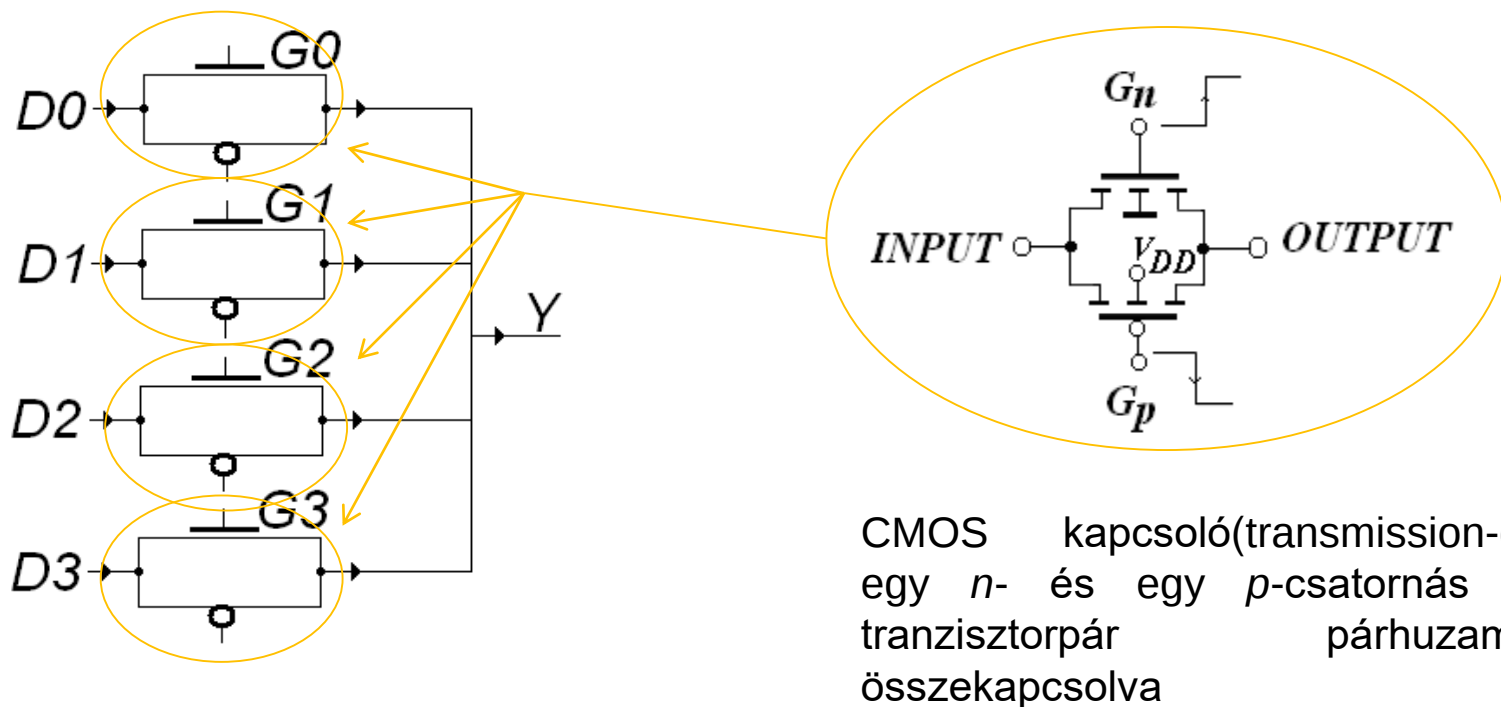
- *A demultiplexerek, mint dekóderek alkalmazása:*
  - ha egy demultiplexer adatbemenetét állandó, logikai '1' szintre kapcsoljuk, akkor ez egyenértékű azzal, hogy a hálózatában szereplő 'ÉS' kapuk bemenetei közül elhagyjuk az adatbemenetet. Az ilyen áramkör sajátossága, hogy a kiválasztó bemenetekre kapcsolt kombinációk csak egyetlen, a kiválasztó kódnak megfelelő kimeneten eredményeznek magas szintet. Az ilyen áramköröket nevezzük dekódereknek.

## Demultiplexerek

- Példa DMPX1-4 egység dekóderként történő felhasználására:*



## Az átvivőkapu (transmission gate) alkalmazása a multiplexerekben és a demultiplexerekben



- *A harmadik logikai állapot, a „lebegő” kimenet lehetősége!*

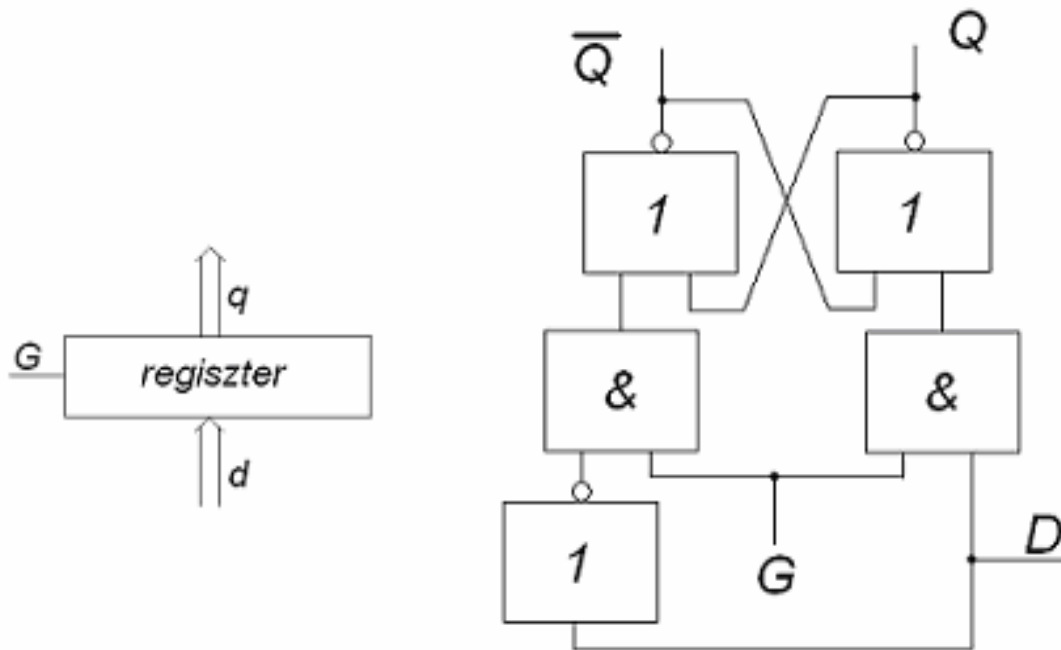
## Regiszterek

- *adatokat tárolnak, és ezek elérését biztosítják*

### *Alapelemek:*

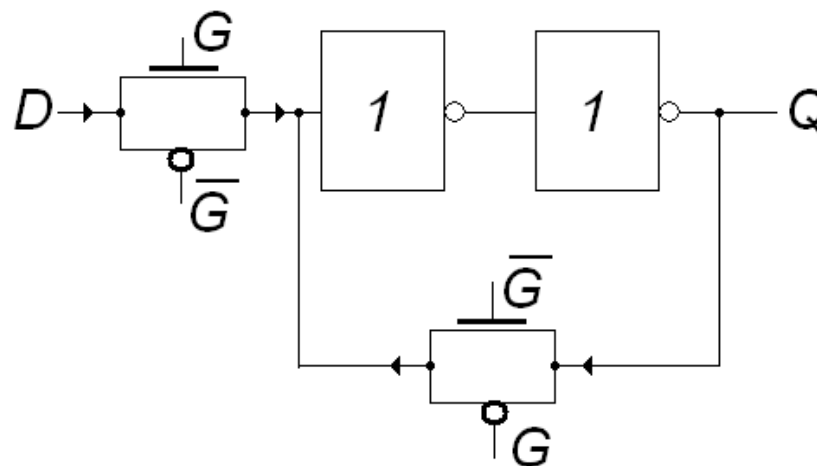
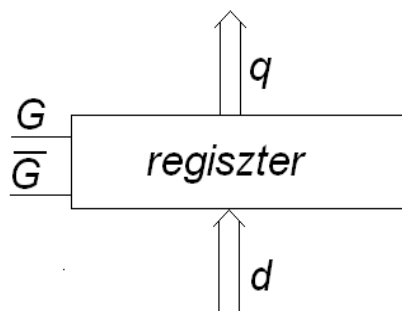
- *szintvezérelt statikus regiszter*
- *szintvezérelt statikus regiszter ponált és negált beírójelekkel*
- *kvázistatikus regiszter*
- *élvezérelt regiszter*

## Szintvezérelt statikus regiszter



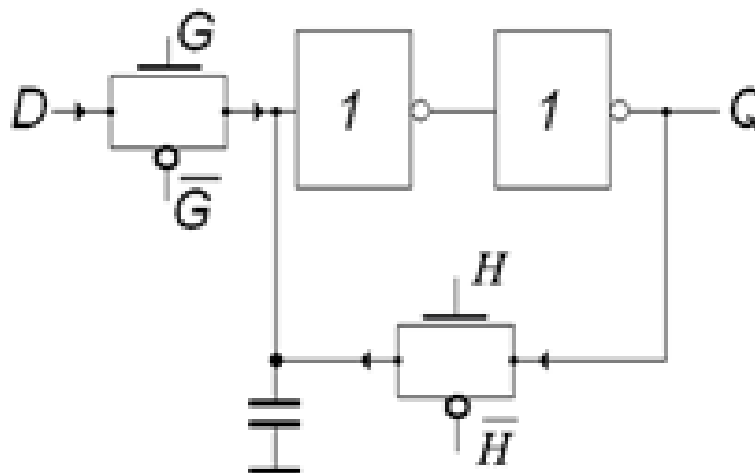
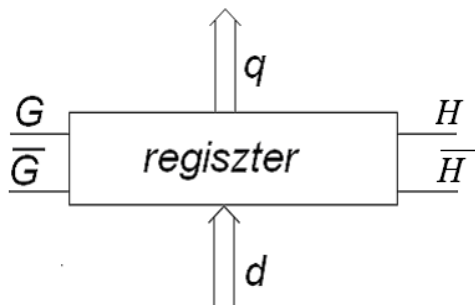
- A regiszter a  $G=1$  szint fennállásának idején „átlátszó”, azaz  $d$  változásai késleltetve ugyan, de kijutnak a kimenetre.

## Szintvezérelt statikus regiszter ponált és negált beírójelekkel



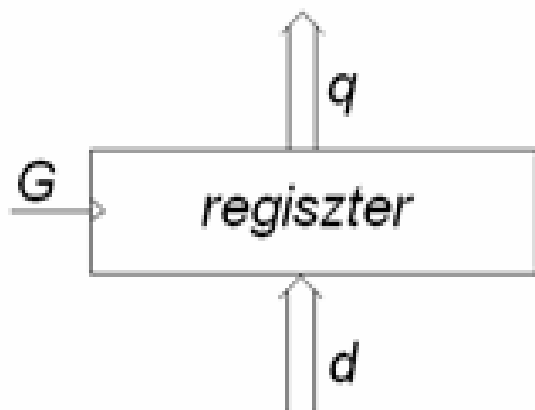
- CMOS átvivőkapu alkalmazása statikus regiszterben

## Kvázistatikus regiszter



- A kapacitás a **G** lefutása és **H** felfutása között tárolja a beírt szintet, az inverterek „frissítenek”.

## Élvezérelt regiszter



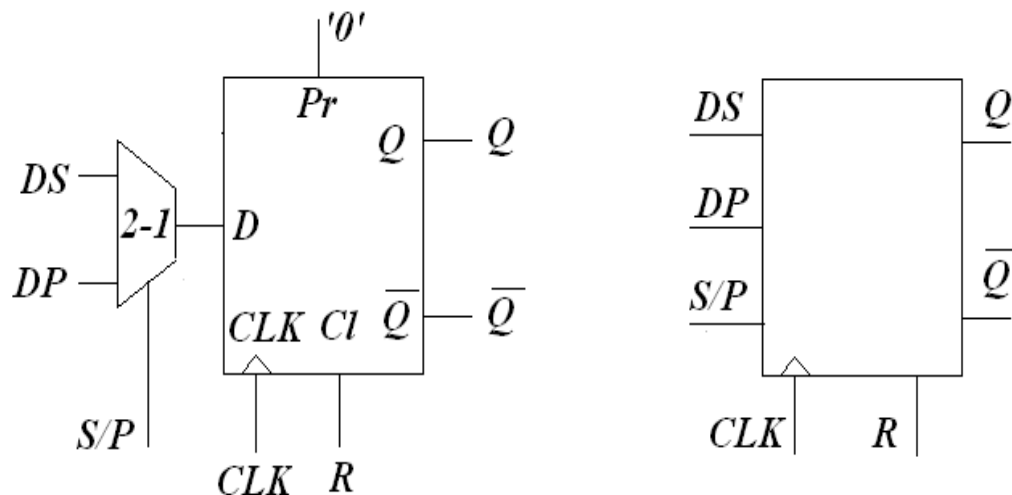
- Az átlátszóság a **G** jel felfutásának idejére szűkül! Igen sok előny származik ebből.

## Tárolók

- soros elérésű memóriák
- párhuzamos hozzáférésű memóriák

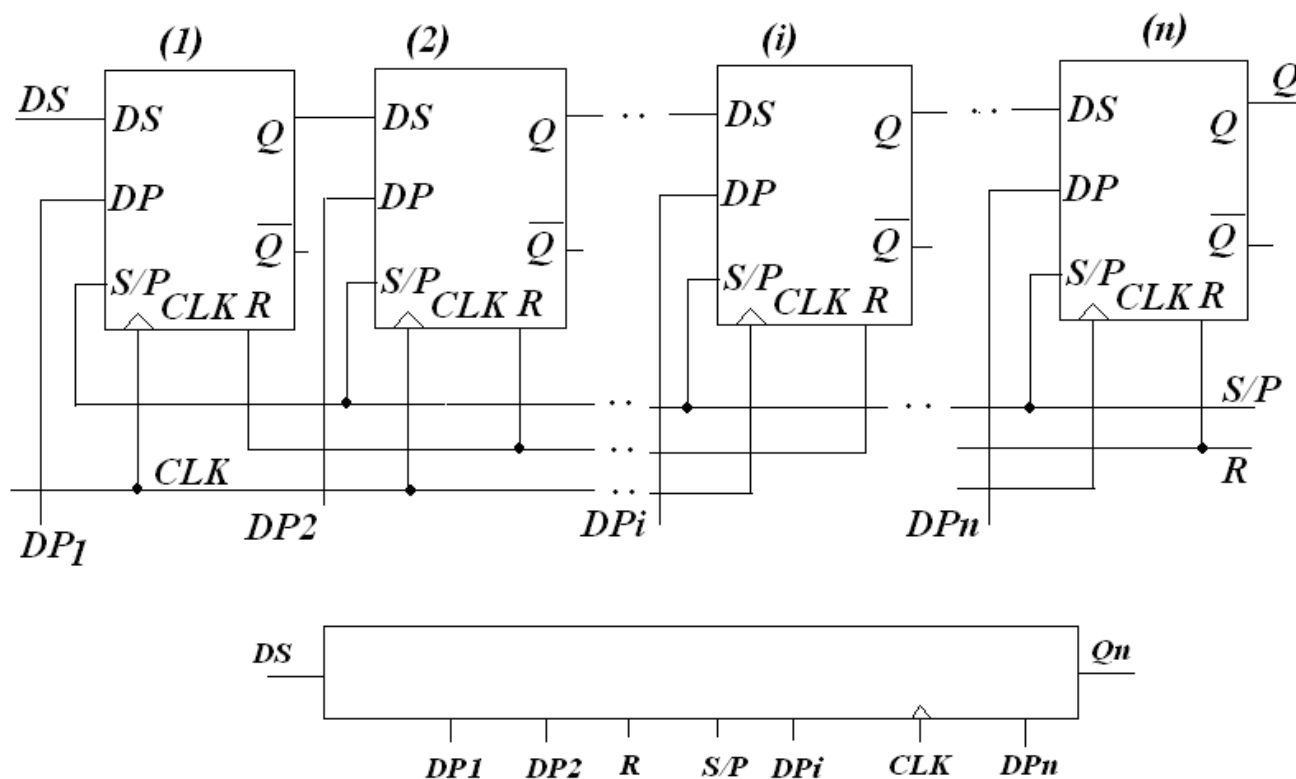
## Soros elérésű memóriák

A soros memóriák alapeleme:

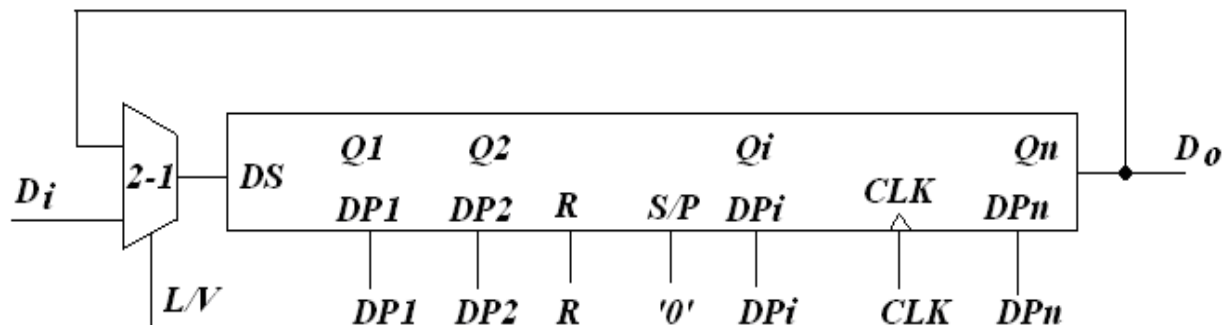


- két bemenetről ( $DS$ ;  $DP$ ) beírható élvezérelt D-MS flip-flop, a  $D$  bemeneten MPX2-1 multiplexerrel.

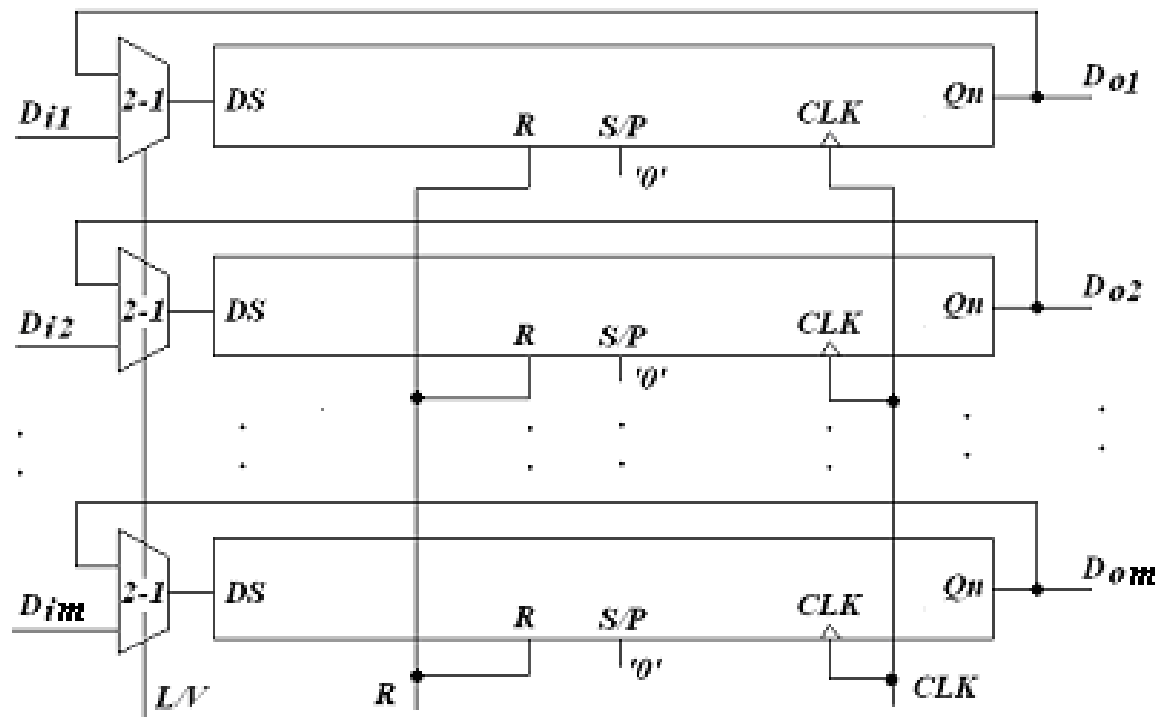
## Nyitott, párhuzamosan is betölthető soros elérésű memória-sor (SHIFT-regiszter)



## Bit-szervezésű, sorosan rátölthető, párhuzamosan is betölthető soros elérésű memória

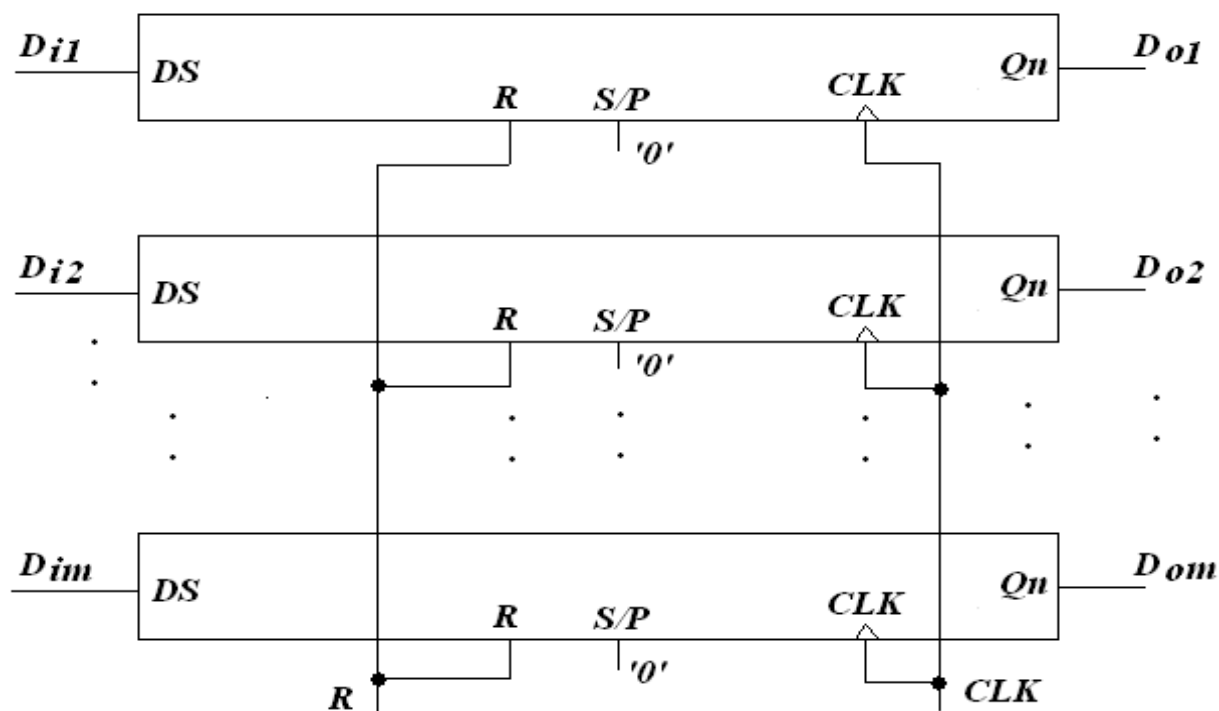


## Szószervezésű soros memóriák (1): sorosan rátölthető soros elérésű memória



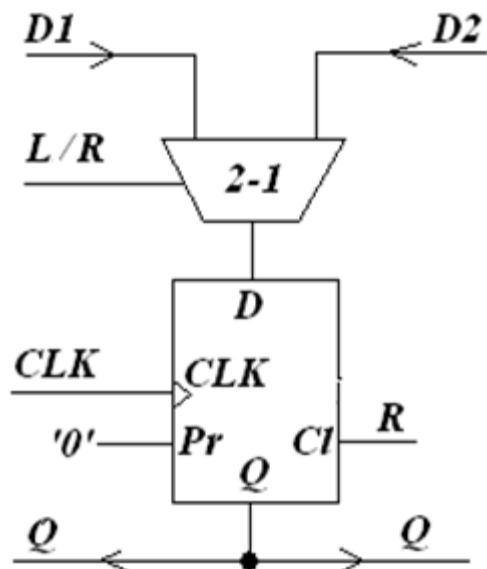
- $m$ -bit szélességű soros memóriaegység

## Szószervezésű soros memóriák (2): F-I-F-O memóriák (First In First Out)

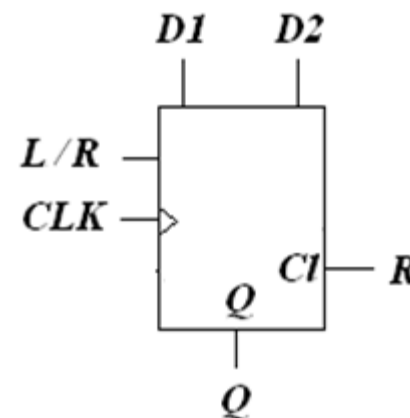


- $m$ -bit szélességű FIFO memória

## Szószervezésű soros memóriák (3): L-I-F-O memóriák (Last In First Out)

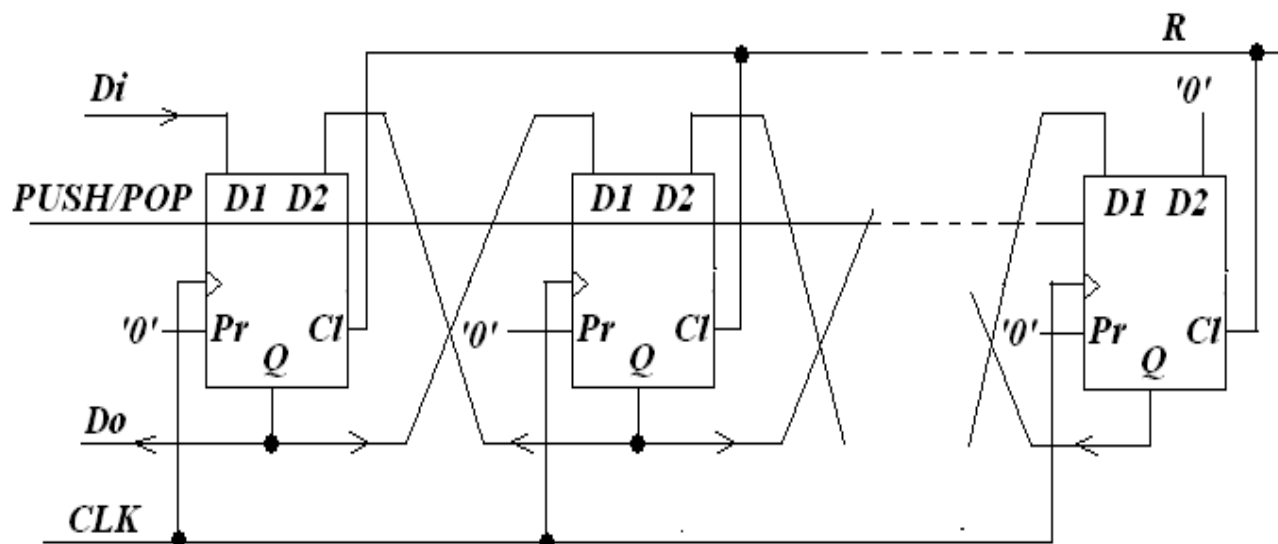


sematikus  
ábra:



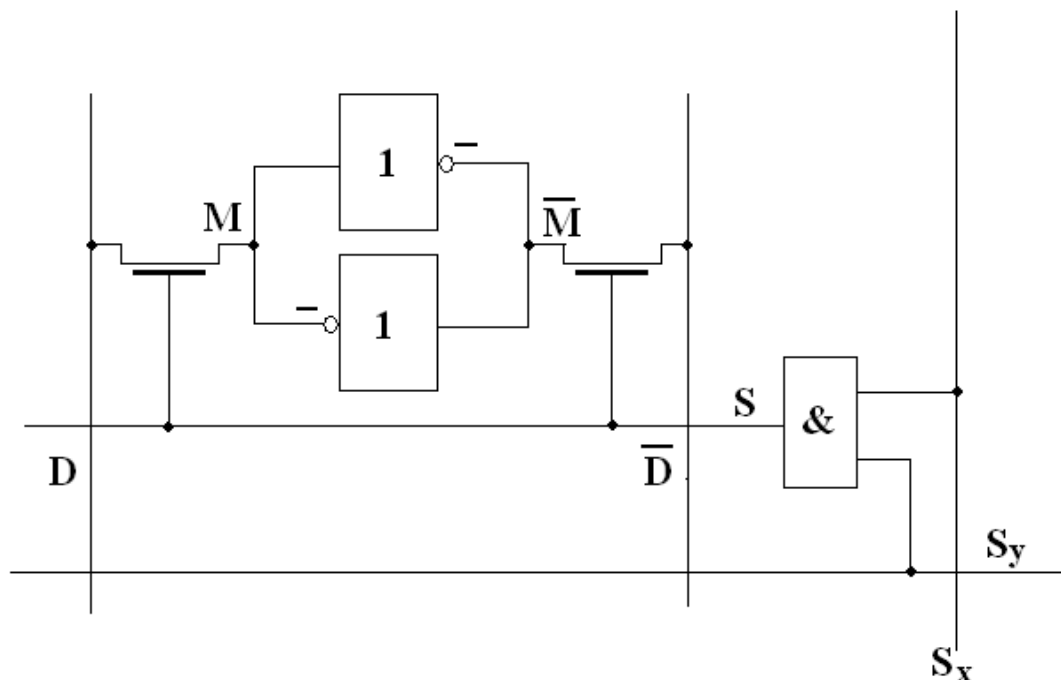
- LIFO memóriaelem

## Szószervezésű soros memóriák (3): L-I-F-O memóriák (Last In First Out)



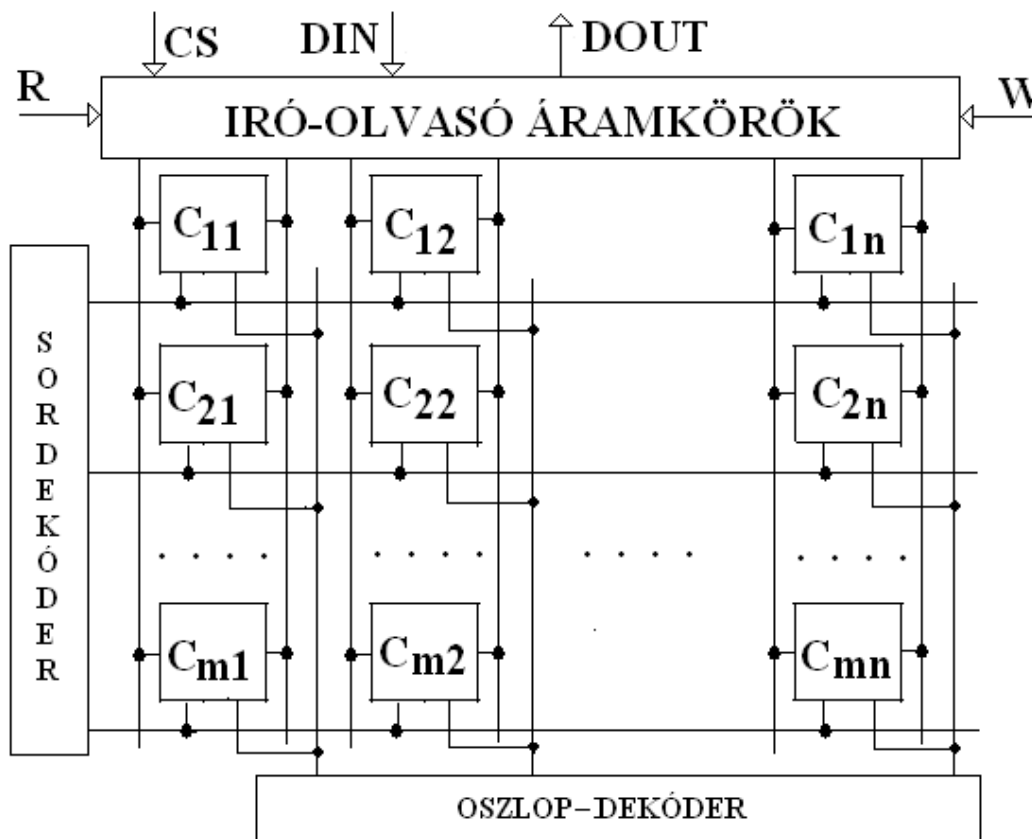
- LIFO memóriasor

## Párhuzamos hozzáférésű memóriák (RAM-ok) (1)



- RAM alapcella felépítése

## Párhuzamos hozzáférésű memóriák (RAM-ok) (2)

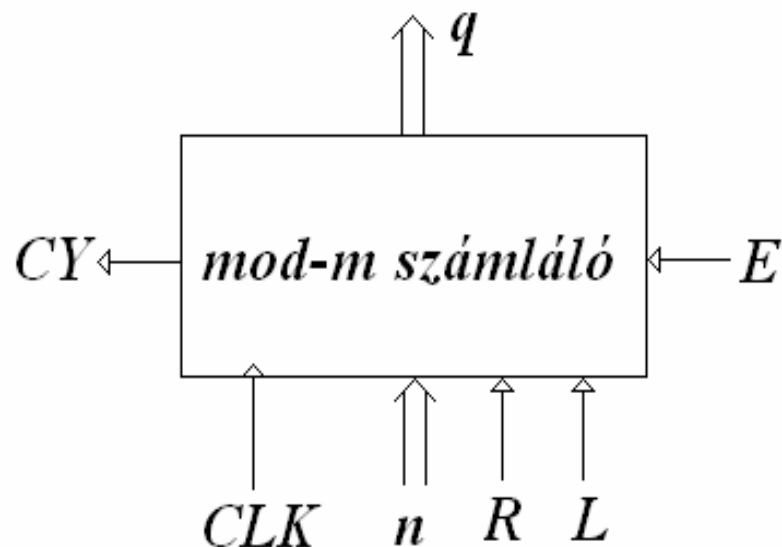


- Bit szervezésű RAM blokk

## Állapotregiszterek, számlálók

- szinkron számlálók
- aszinkron számlálók

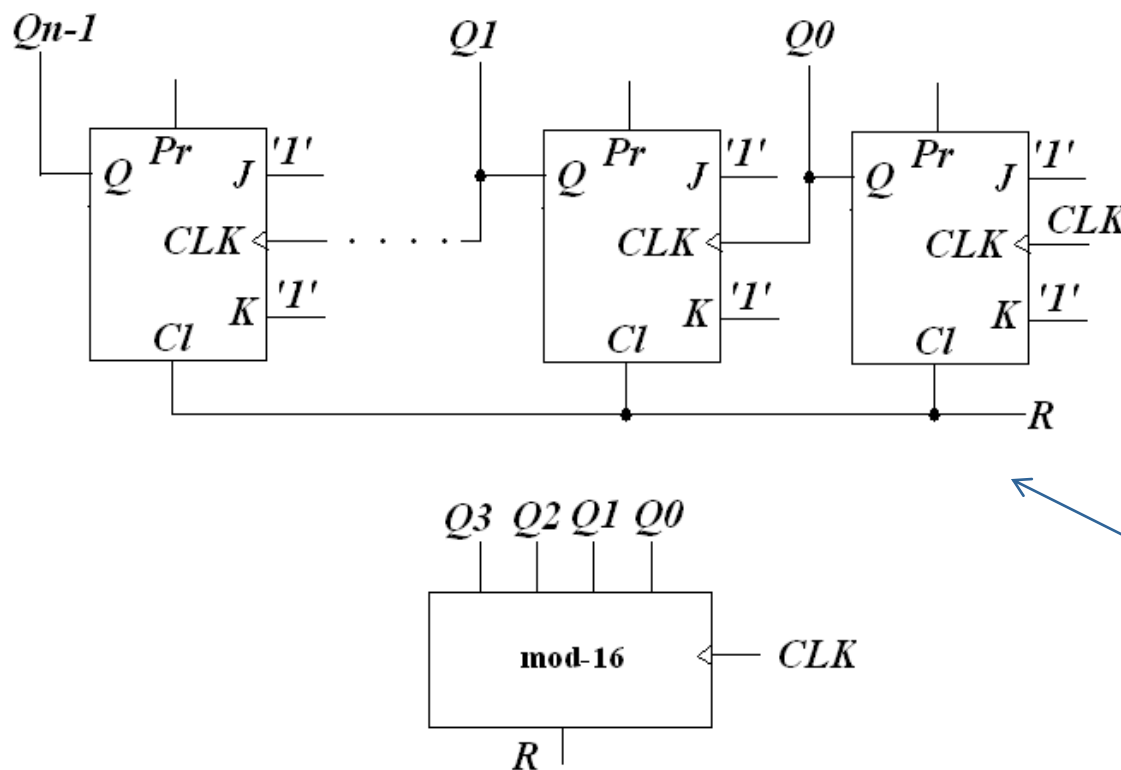
## *Szinkron számlálók (ism.)*



- *mod-m* szinkron számláló általános sémája

(...)

## Aszinkron számlálók



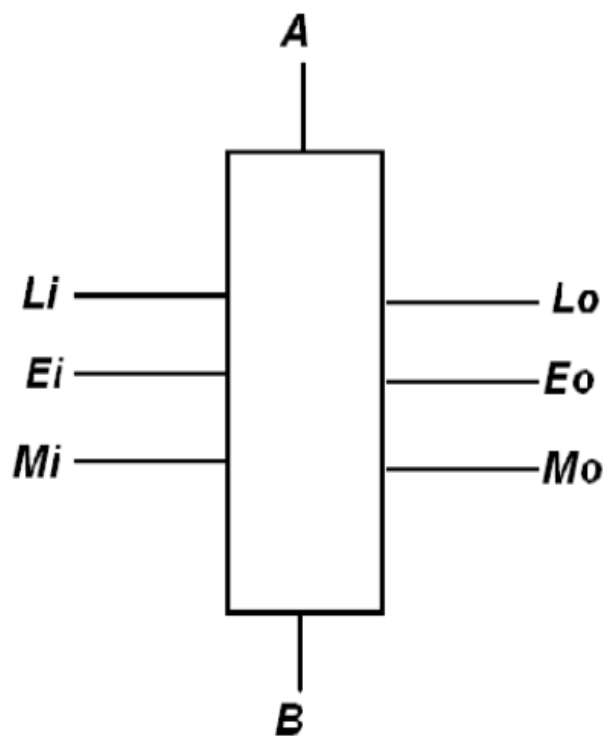
- Kettes osztók kaszkádja

## **Funkciós egységek**

- komparátorok
- összeadók, kivonók
- szorzók, osztók
- vezérlők
- ...

## Komparátorok

*Az 1-bites komparátor*



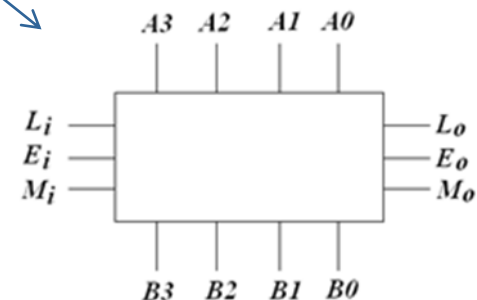
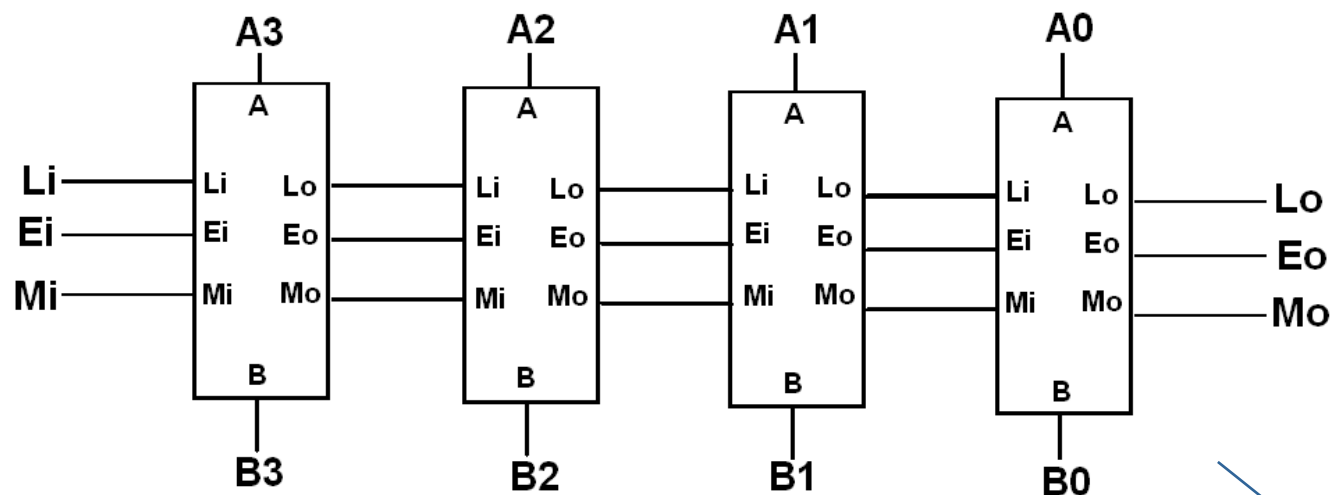
$$L_o = L_i + E_i (\bar{A} \bar{B})$$

$$E_o = E_i (\bar{A} \bar{B} + A B)$$

$$M_o = M_i + E_i (A \bar{B})$$

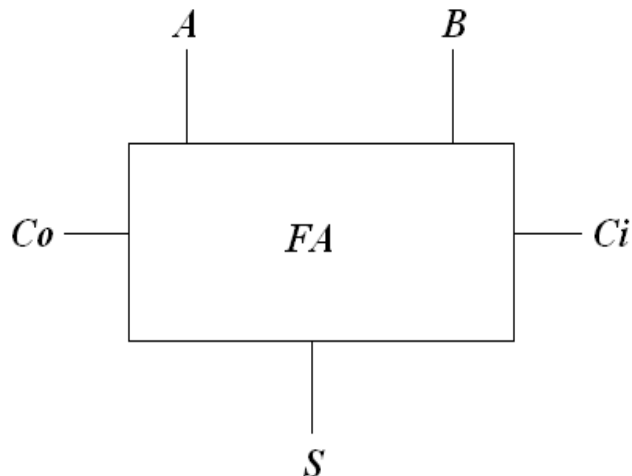
## Komparátorok

Példa: 4-bites komparátor kialakítása 1-bites alapegységekből



## Összeadók

### Az 1-bites (teljes) összeadó



- A teljes összeadó sémája

- igazságtábla

| A | B | C <sub>i</sub> | S | C <sub>o</sub> |
|---|---|----------------|---|----------------|
| 0 | 0 | 0              | 0 | 0              |
| 0 | 0 | 1              | 1 | 0              |
| 0 | 1 | 0              | 1 | 0              |
| 0 | 1 | 1              | 0 | 1              |
| 1 | 0 | 0              | 1 | 0              |
| 1 | 0 | 1              | 0 | 1              |
| 1 | 1 | 0              | 0 | 1              |
| 1 | 1 | 1              | 1 | 1              |



$$S = (A \oplus B) \oplus C$$

$$C_o = A B + B C_i + A C_i$$

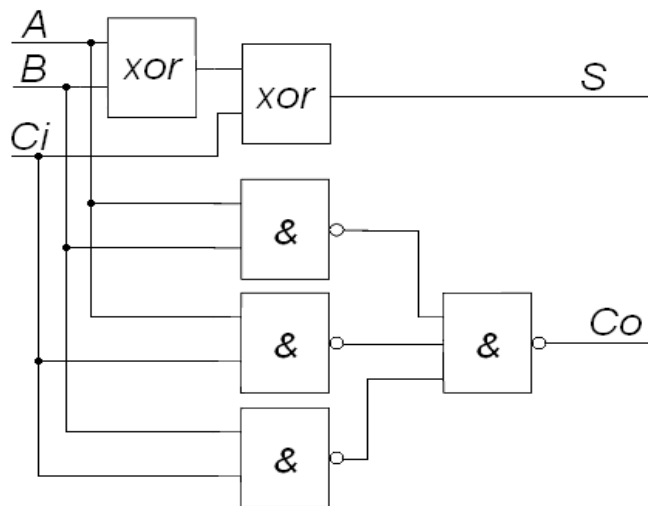
## Összeadók

### Az 1-bites (teljes) összeadó

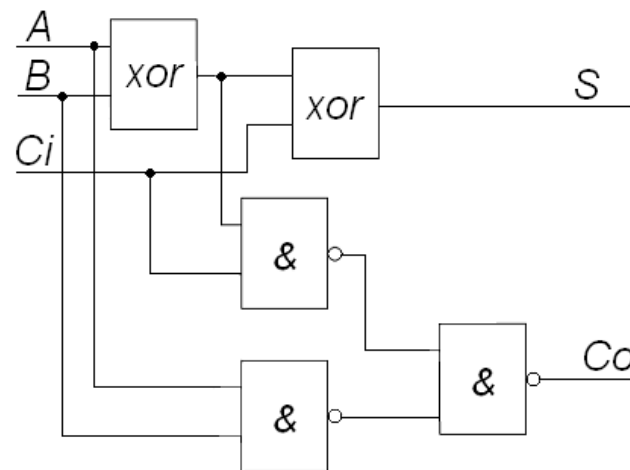
$$S = (A \oplus B) \oplus C$$

$$C_o = A B + B C_i + A C_i = \overline{(AB)} \overline{(B C_i)} \overline{(A C_i)} \quad : (1)$$

$$C_o = (A \oplus B) C_i + A B = \overline{(A \oplus B) C_i} \overline{(AB)} \quad : (2)$$



:(1)

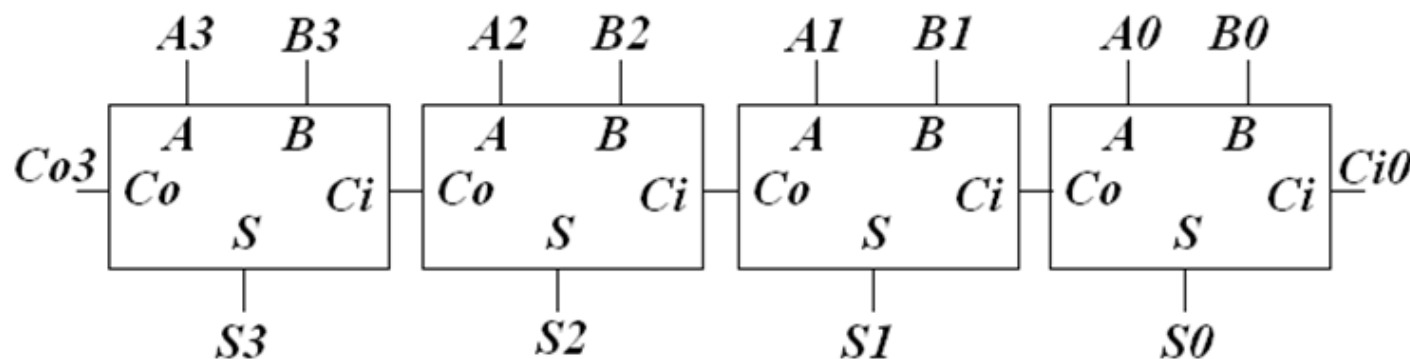


:(2)

## Összeadók

### *Soros átvitelképzésű összeadó*

Példa: soros átvitelképzésű 4-bites összeadó egység, 1-bites alapegységek kaszkádosításával



## Összeadók

### *Párhuzamos átvitelképzésű bit-vektor összeadó*

- eredményképzés algoritmus:

$$P_k = A_k \oplus B_k$$

$$G_k = A_k B_k$$

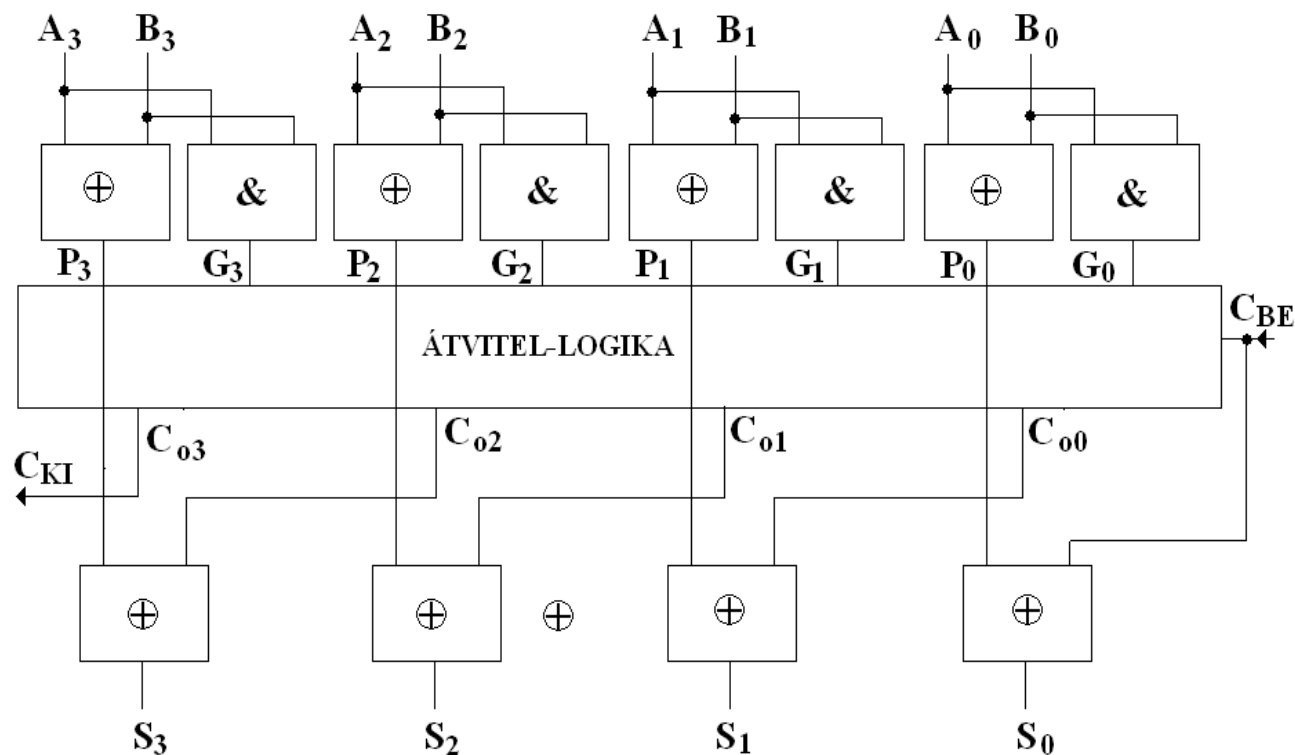
$$S_k = (A_k \oplus B_k) \oplus C_{ik} = P_k \oplus C_{o(k-1)}$$

$$C_{ok} = (A_k \oplus B_k) C_{ik} + A_k B_k = P_k C_{o(k-1)} + G_k$$

## Összeadók

### *Párhuzamos átvitelképzésű bit-vektor összeadó*

Példa: 4-bites, párhuzamos átvitelképzésű összeadó



## Kivonók

### A komplement kód alkalmazása a kivonás műveletében

- elv: a kivonás műveletét a bináris számok kódolásának megfelelő megválasztásával összeadásra lehet visszavezetni.
- megoldás: vegyük a kivonandó kettes komplementjét, és a kisebbítendőhöz adjuk hozzá!

## A kettes-komplementens kódú számábrázolás

**A**: szám, súlyozott bináris kóddal

**$KK(A)$**  : a szám kettes komplementense, adott szabály szerint előállítva.

Egy kettes komplementens kódú szám  $(-1)$ szerese a szám kettes komplementense:

$$A + KK(A) = 0!$$

*A kettes komplementens kód:*

**MSB** : előjel

**$(MSB-1) - LSB$**  : számérték

- *ha a szám pozitív, előjele 0, a számérték pedig a szám binárisan súlyozott abszolút értéke*
- *ha a szám negatív, előjele 1, és az abszolút érték a kettes komplementens előállításával határozható meg*

## A kettes komplementes előállítás:

1. lépés : bitenkénti negálás (egyes komplementes)
2. lépés :  $000\dots1$  hozzáadása az egyes komplementeshez

**Példa:** 0 1 0 1 pozitív szám, abszolút értéke 5,  
ez tehát a (+5) kettes komplementes kóddal.

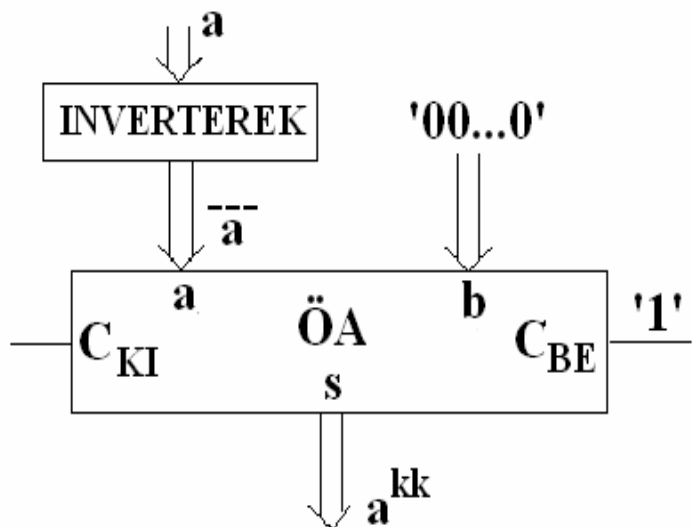
Ennek 2-es komplementese -5 kell hogy legyen:

1-es komplementes : 1 0 1 0

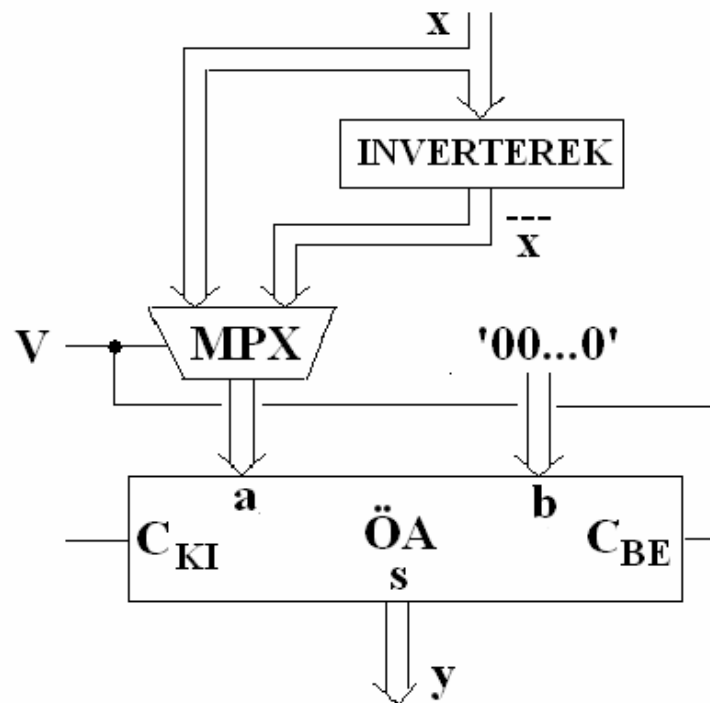
2-es komplementes : 1 0 1 1

Próba : 0 1 0 1  
+ 1 0 1 1  
-----  
(1)0 0 0 0

## Kivonók megvalósítása (1) kettes-komplementens képző egységek

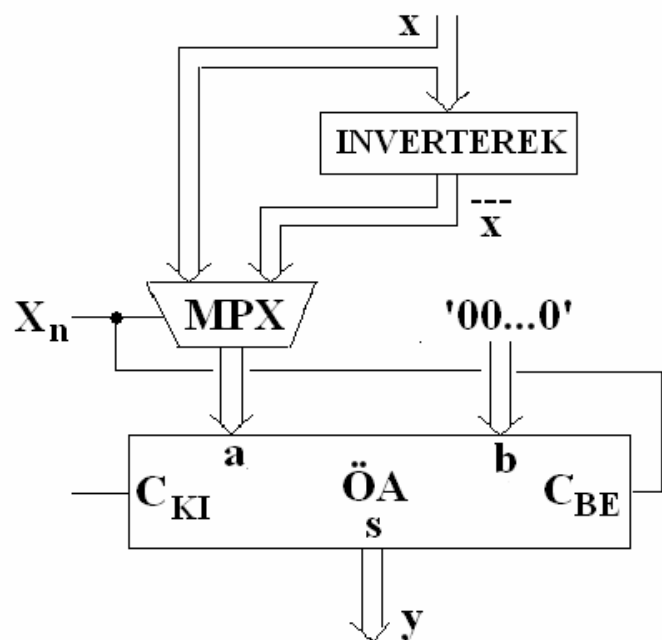


- *kettes-komplementens képző egység*

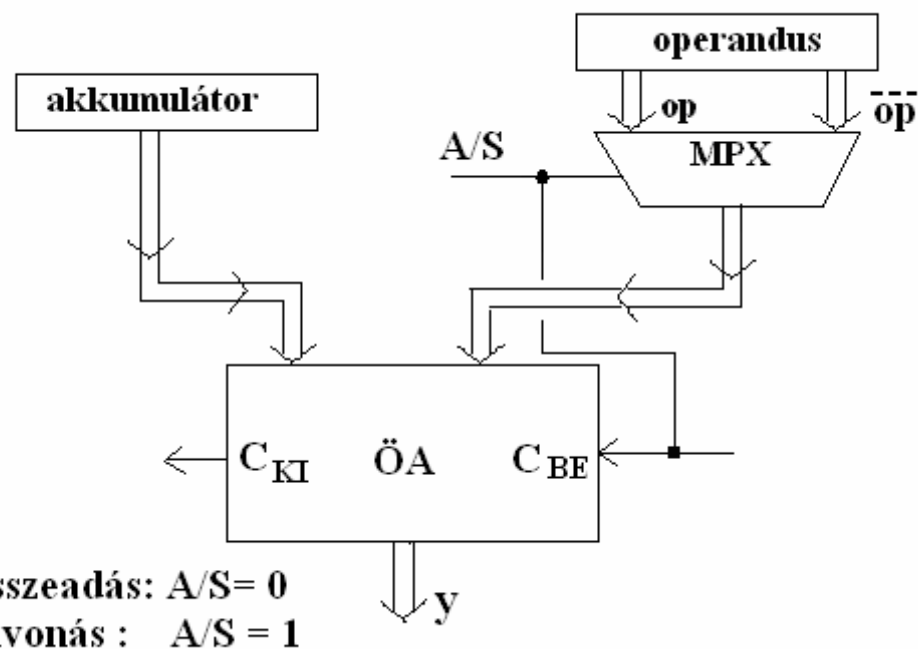


- *vezérelhető kettes-komplementens képző egység*

## Kivonók megvalósítása (2)



- abszolútérték képző egység



- kivonás mikroprocesszorok aritmetikai egységében

## Szorzók, osztók

➤ *különböző elvek → sokféle megvalósítás*

*néhány példa:*

- szorzók:
  - előjeles vagy előjel nélküli szekvenciális szorzók
  - tömbszorzók (array-szorzók)
  - fa-szorzók (Booth-Wallace szorzók)
  - ...
- osztók:
  - előjel nélküli visszaállító osztó
  - előjeles nem visszaállító osztó
  - Radix 4 SRT osztó
  - funkcionális iterációs osztók (Newton-Raphson algoritmus)
  - ...

## Példa1:

elv:

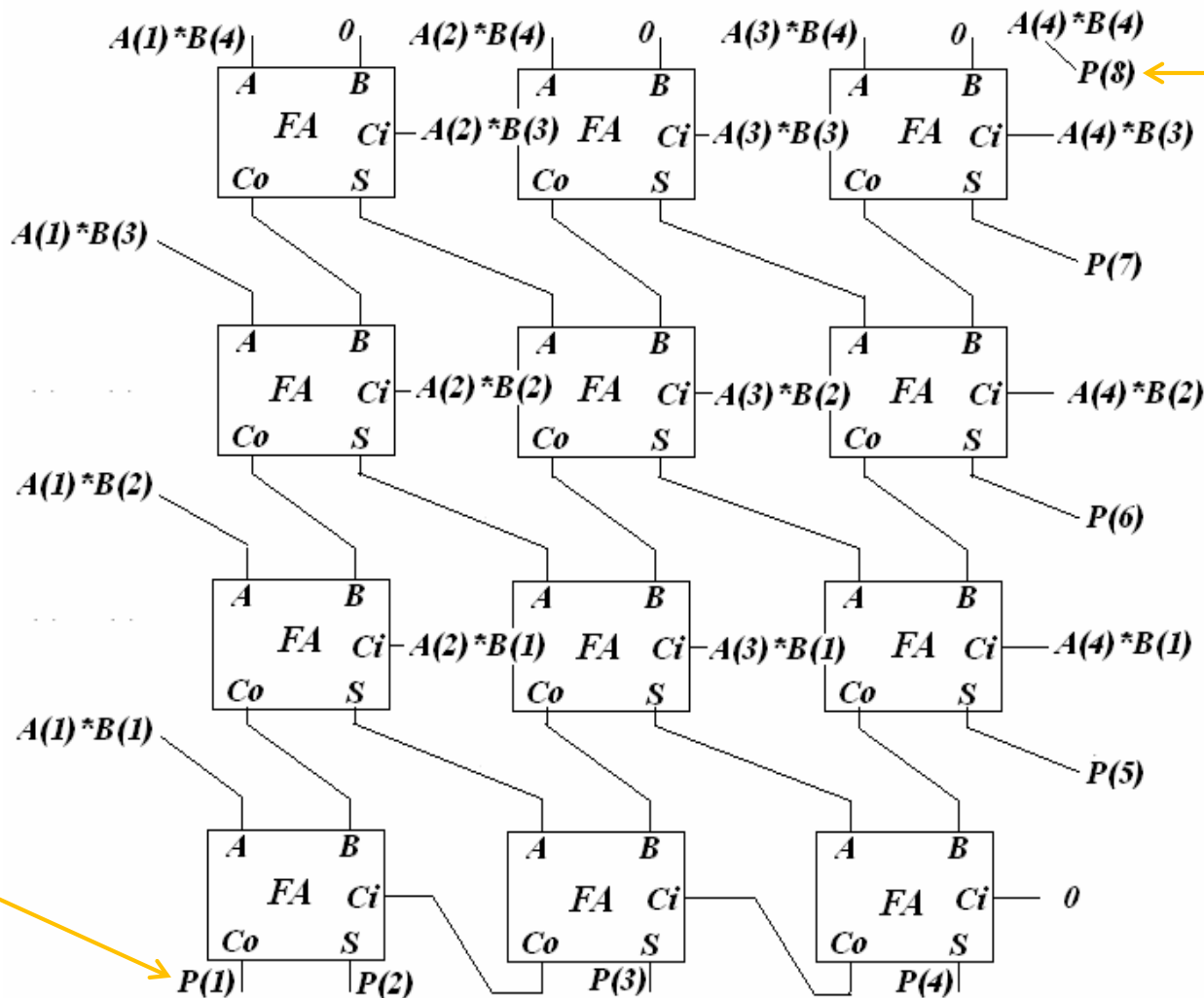
bitenkénti  
**szorzás**



**léptetés**

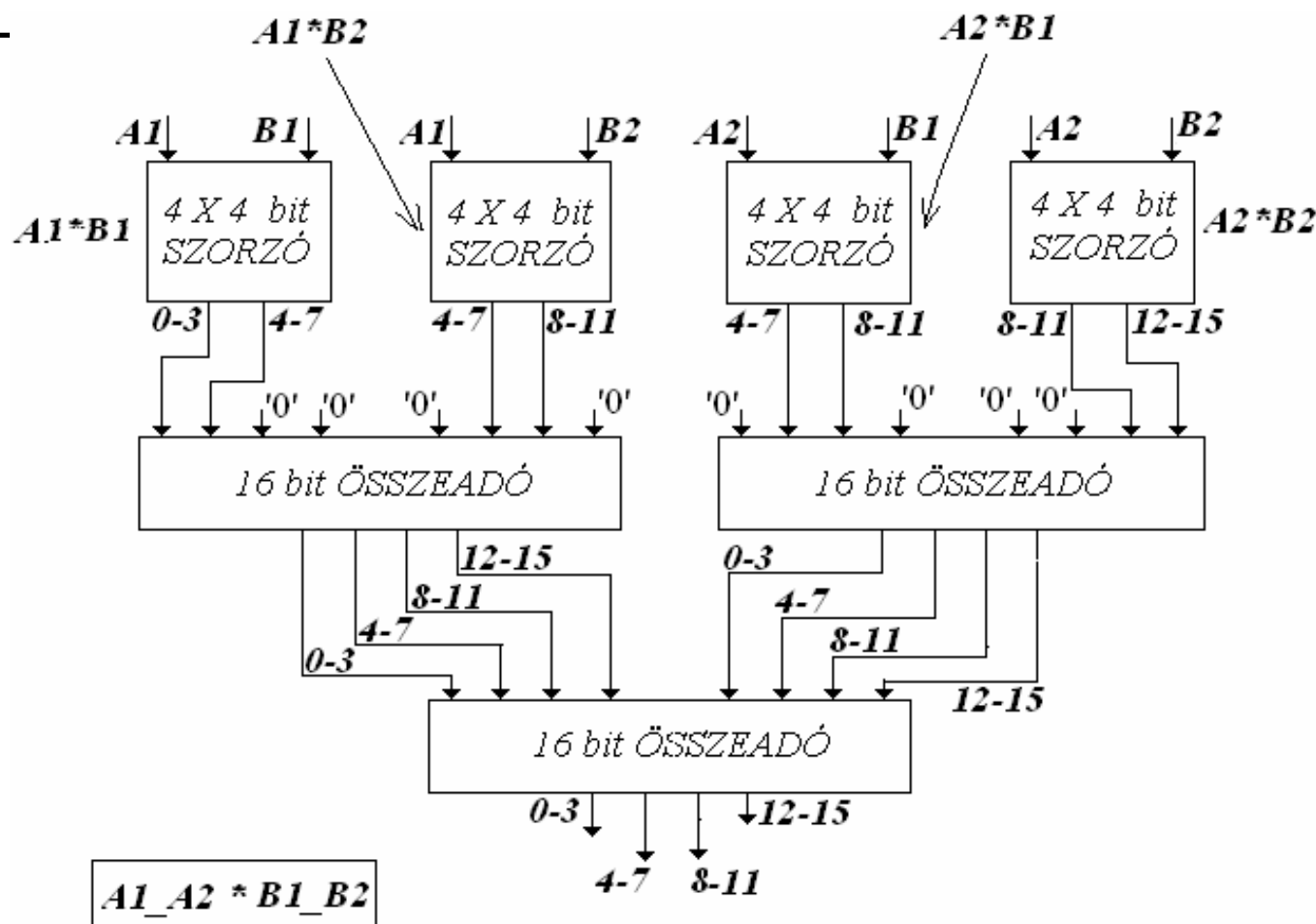


**összeadás**  
műveletsor



- 4 x 4-es tömbszorzó kialakítása teljes összeadókából valamint 'ÉS' kapukból

## Példa2:

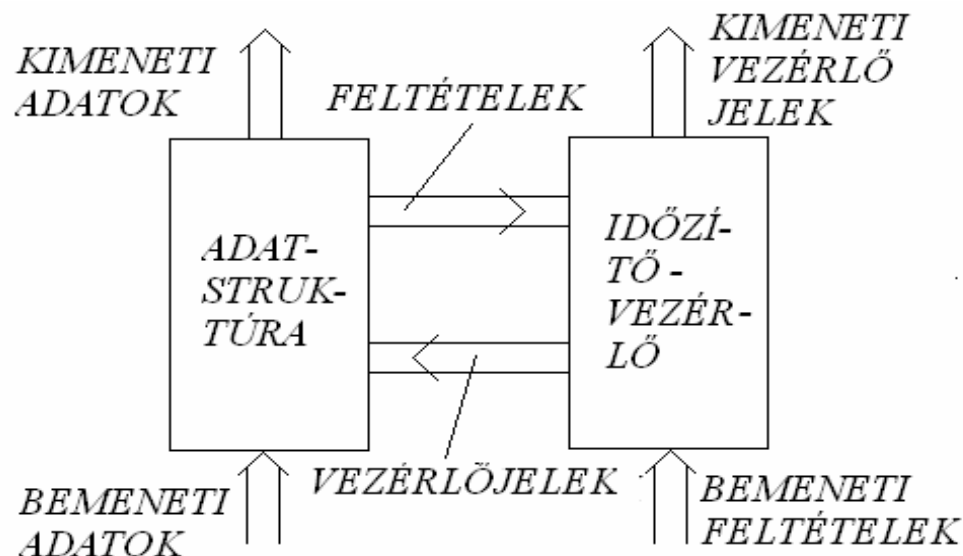


- 8-bites szorzó 4-bites egységekből

## Vezérlők

➤ *Elv: dekompozíció (ism.)*

- elkülönítjük az adatutakat az azok kijelölését és időbeli mozgását meghatározó időzítő vezérlő egységtől:

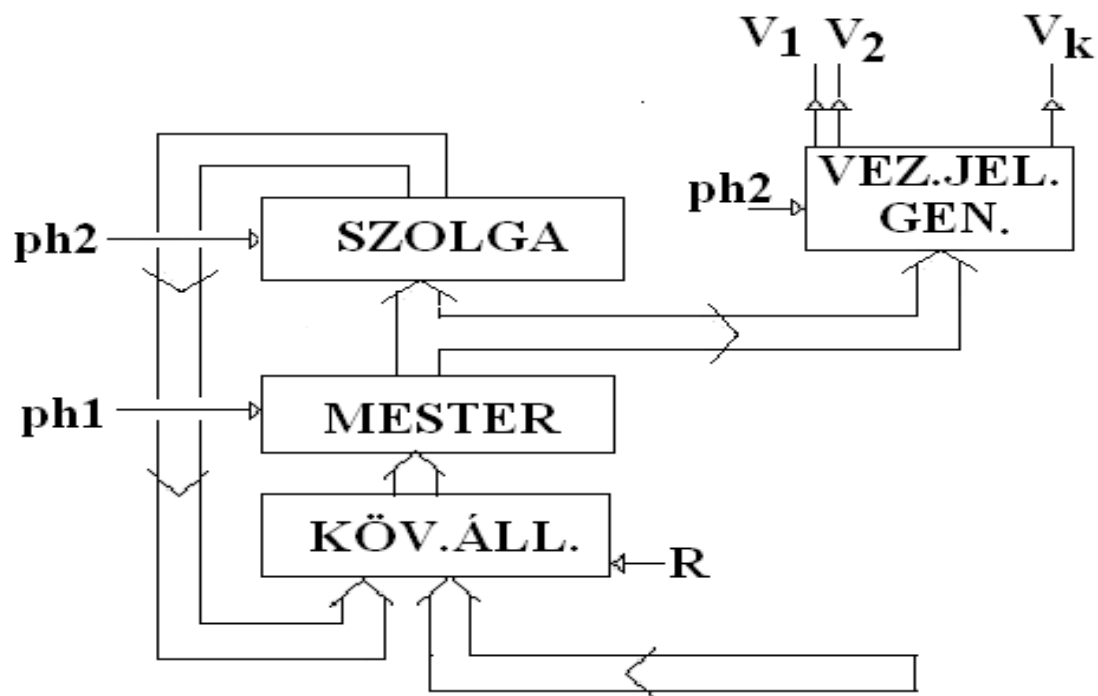


## Vezérlők (1)

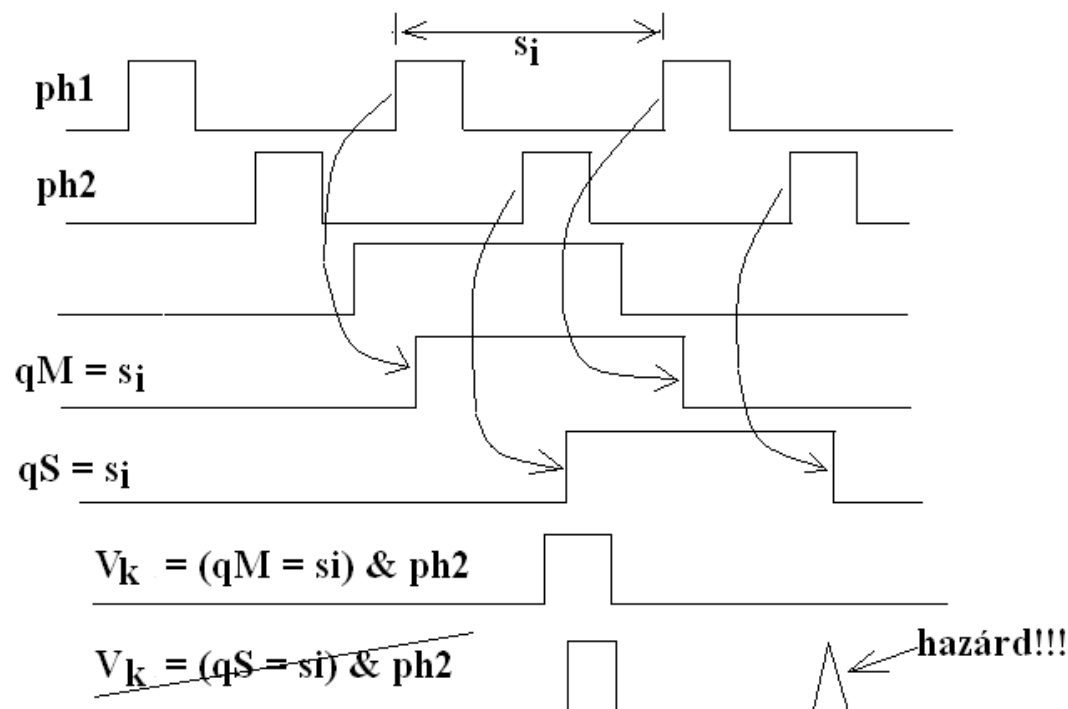
- *számláló bázisú időzítő-vezérlők (ism.)*
- *FSM típusú vezérlők*
  - **Finite State Machine:** az időzítő véges állapotszámú, MS tárolókból az adott célra felépített szinkron sorrendi hálózat.

## Vezérlők (2)

Kétfázisú ( $ph_1$ ,  $ph_2$ ) órajellel működő FSM vezérlő struktúrája:



## Vezérlők (3)



- kétfázisú órajellel működtetett FSM típusú időzítő idő-diagramja