

MISKOLCI EGYETEM

Villamosmérnöki Intézet

Automatizálási Tanszék

DIGITÁLIS ÁRAMKÖRÖK SZIMULÁCIÓJA

Oktatási segédlet (javított és bővített kiadás)

Gépész informatikus, anyagmérnök automatizálási, gépész mechatronikai, földtudomány levegő tisztaság védelem szakirányos és villamos hallgatók részére

<http://mazzola.iit.uni-miskolc.hu/~gardus>

Dr. Gárdus Zoltán Ph.D.

egyetemi adjunktus

Miskolc

2005.

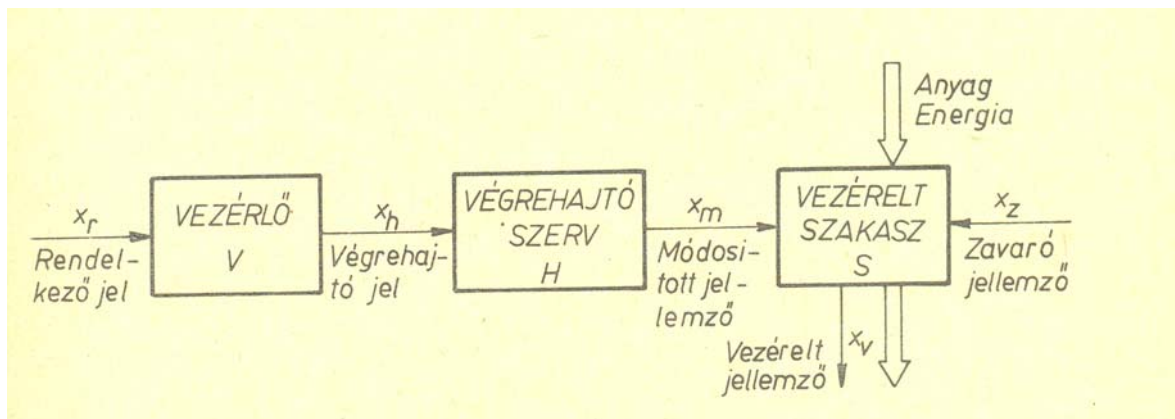
Tartalomjegyzék

1. Vezérléstechnika.....	3
2. Bevezetés a logikai tervezés alapjaiba.....	4
3. Több változós logikai függvények megadási módszerei.....	12
4. Logikai függvények egyszerűsítése, minimalizálása.....	21
5. Kódolási alapfogalmak.....	30
6. Alapkódok bemutatása és képzési szabályuk.....	32
7. Digitális áramkörök.....	42
8. Kombinációs típusú funkcionális egységek.....	44
9. Tárolóelemek, F.F.-ok.....	49
10. Léptetőregiszterek.....	55
11. Hazárd, versenyhelyzet.....	56
12. Aszinkron számlálók felépítése, készítésük és kialakításuk.....	57
13. Szinkron számlálók tervezése, felépítése.....	60
14. Aszinkron sorrendi (szekvenciális) hálózatok.....	69
15. Az alábbiakban közölt feladatsor, az I. minta zárthelyi a III. évfolyam informatikus hallgatói részére.....	81
16. Kidolgozott, szimulációra alkalmas hálózatok.....	83

Az alábbi anyag szorosan kapcsolódik az előadások témaköreihez, melyek megértéséhez elengedhetetlen az előadások látogatottsága

1. Vezérléstechnika

A vezérlés hatáslánca nyílt, azaz a folyamat kimenetéről visszacsatolással nem rendelkező irányítási művelet. A vezérlés művelete úgy zajlik le, hogy egy vagy több rendelkezés hatására a vezérlő berendezés végrehajtó jelei meghatározott törvényszerűséggel működtetik a végrehajtó szerveket. Az utóbbiak a módosított jelek által hatnak a vezérelt szakaszban lezajló műszaki folyamatokra.



A fenti ábra a vezérlés hatáslancát szemlélteti.

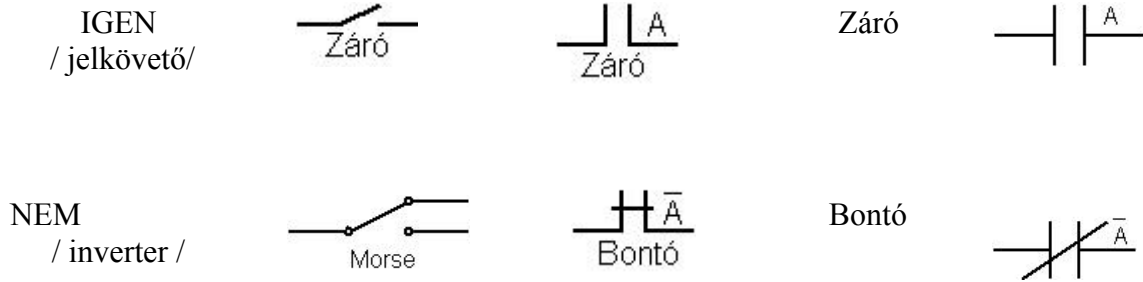
A vezérlő berendezés lehetséges alternatívái, a kronológiai sorrendet figyelembe véve az alábbiak lehetnek:

- érintkezős, relés logikák;
- digitális (TTL, CMOS) áramkörökkel kialakított huzalozott vezérlések;
- PLA, FPLA, PAL, GAL áramkörökkel realizált hálózatok;
- PLC-s, PV-s és PC-s (Programmable Controller) irányítások;
- célorientált DSP-vel, ill. mikrokontrollerekkel működtetett folyamatok;
- mikroprocesszorokkal felépített irányítási rendszerek.

2. Bevezetés a logikai tervezés alapjaiba

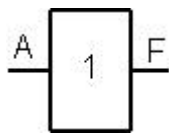
Érintkező szimbólumok:

PLC-knél létradiagramos programozás módszer esetén



Egyváltozós logikai alapfüggvények

IGEN, JELKÖVETŐ, JELMÁSOLÓ, YES kapcsolat MSZ jelképe



2^n
kombinációs
lehetőség

n: bemenetek számát jelöli

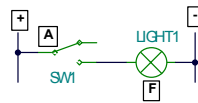
Kombinációs tábla

KV- tábla

Érintkezős realizáció

mi	A	F
0	0	0
1	1	1

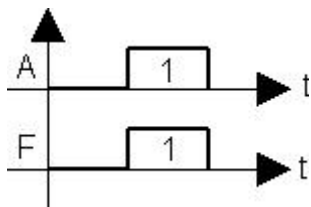
A	0
	1 ₁



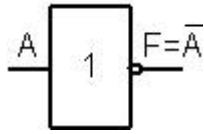
Indexszámok alak

$$F^1_2(A) = 0.2^0 + 1.2^1$$

Idődiagram



NEM, TAGADÁS, INVERZIÓ, NOT, INVERTER MSZ jelképe



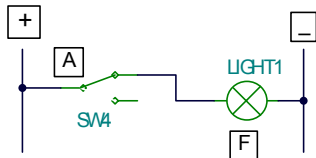
Kombinációs tábla KV -tábla

mi	A	F
0	0	1
1	1	0

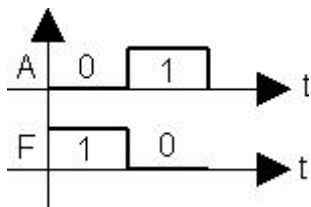
A	1 ₀
	1

$$F_1^1(A) = 1 * 2^0$$

Érintkezős realizáció



Idődiagram



Kétváltozós logikai alapfüggvények

ÉS, AND kapcsolat

(n = 2 a bemenetek minimális számát jelöli)

$$F^2(A,B) = AB = A \cdot B = A \& B = A \wedge B$$

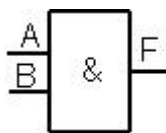
Kombinációs tábla

mi	2 ¹	2 ⁰	F
	A	B	
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

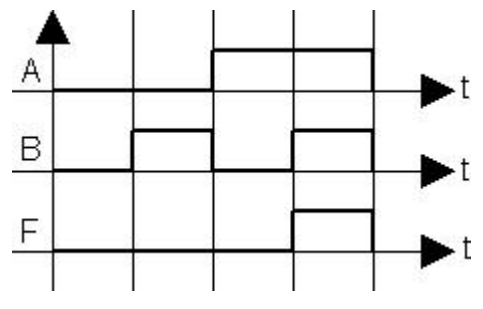
KV-tábla

	B	
A	0	1
	2	3

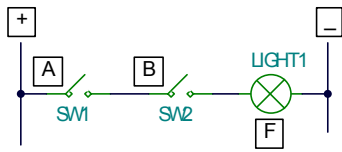
MSZ jelkép



Idődiagram



Az ÉS függvény záró érintkezők soros kapcsolása.



Indexszámú alak

$$F_8^2(A,B) = 1 \cdot 2^3$$

VAGY, OR kapcsolat

$$F^2(A,B) = AVB = A + B$$

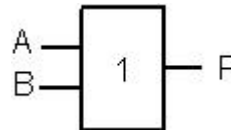
Kombinációs tábla

mi	A	B	F
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

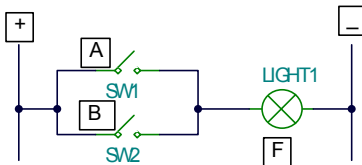
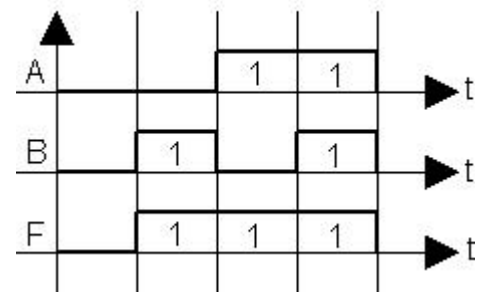
KV tábla

B	
0	1 ₁
1 ₂	1 ₃

MSZ jelkép



Idődiagram



Érintkezős realizáció.

Tehát a VAGY függvény záró érintkezők párhuzamos kapcsolása.

Indexszámú alak:

$$F_{14}^2(A,B) = 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3$$

NEM ÉS, NÉS, NOT AND, NAND kapcsolat

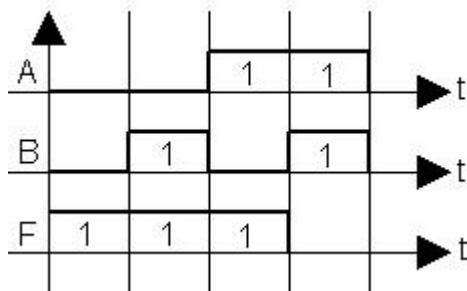
$$F^2(A,B) = \overline{A \& B}$$

A De Morgan azonosságok: $\overline{A + B} = \overline{A} \cdot \overline{B}$ logikai összeg tagadottja egyenlő a bemeneti változók tagadottjának logikai szorzatával. $\overline{A \cdot B} = \overline{A} + \overline{B}$ a logikai szorzat tagadottja egyenlő a bemeneti változók tagadottjának logikai összegével.

Kombinációs tábla

mi	A	B	F
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

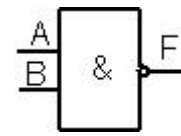
Idődiagram



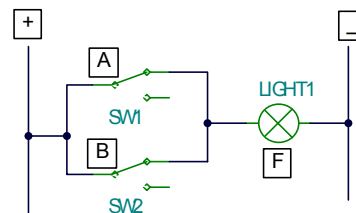
KV-tábla

		B	
A	0	1 ₀	1 ₁
	1	1 ₂	3

MSZ jelkép



Érintkezős realizáció



Indexszámok alak

$$F_7^2(A,B) = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2$$

NEM VAGY, NOT OR, NVAGY, NOR kapcsolat

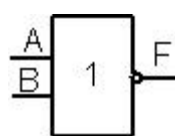
$$F^2(A,B) = \overline{A+B} = \bar{A} \& \bar{B}$$

mi	A	B	F
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0

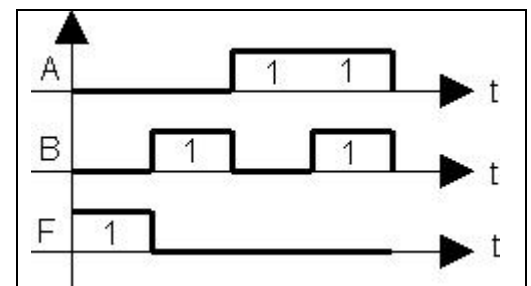
KV tábla

		B	
A	0	1 ₀	1
	1	2	3

MSZ jelkép



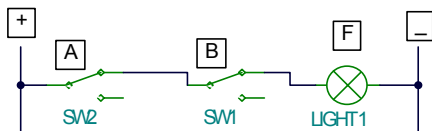
Idődiagram



Indexszámok alak

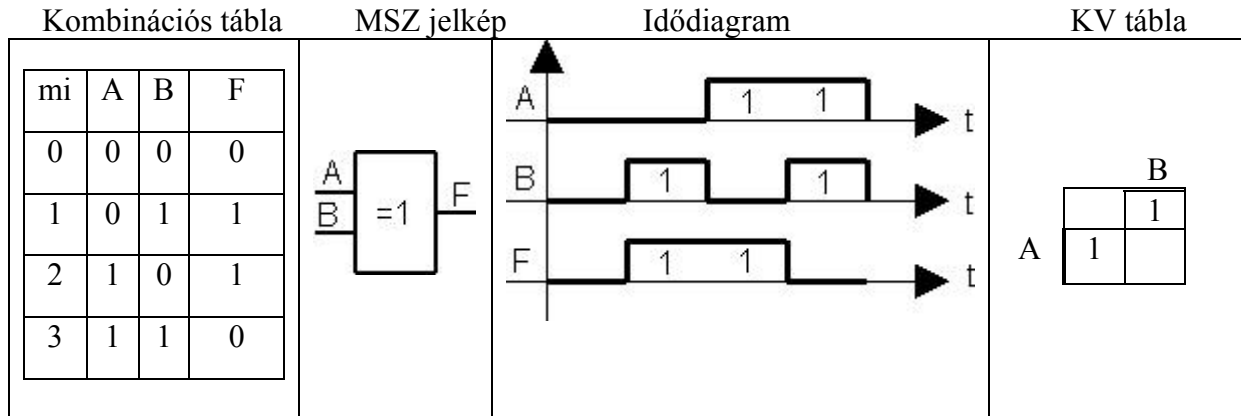
$$F_1^2(A,B) = 1 \cdot 2^0$$

Érintkezős realizáció



ANTIVALENCIA, kizáró VAGY, XOR, EXCLUSIVE OR, EXOR, alternatív kapcsolás (a villanyszerelők által használatos elnevezés)

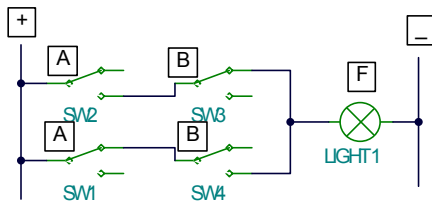
$$F^2(A,B) = A \oplus B = \bar{A}B + A\bar{B}$$



Indexszámú alak

$$F_6^2(A,B) = 1 \cdot 2^0 + 1 \cdot 2^2$$

Érintkezős realizáció



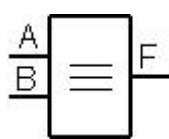
EKVIVALENCIA, EXCLUSIVE NOR, EXNOR, azonosság

$$F_2^2(A,B) = A \oplus B = AB + \overline{A}\overline{B}$$

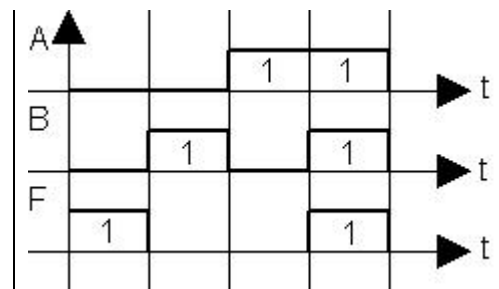
mi	A	B	F
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

		B
A	1	1

MSZ jelkép



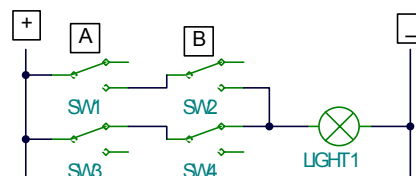
Idődiagram



Indexszámú alak

$$F_9^2(A,B) = 1 \cdot 2^0 + 1 \cdot 2^3$$

Érintkezős realizáció



INHIBÍCIÓ (tiltás), nem kommutatív logikai művelet

$$F^2_4(A,B) = A \& \bar{B}$$

IMPLIKÁCIÓ, nem kommutatív logikai művelet

$$F^2_{11}(A,B) = \bar{A} + B$$

Minden függvényhez tartozik egy másik **duális** függvény, amely csak abban tér el, hogy benne az AND és az OR műveletek fel vannak cserélve.

Két függvény akkor **inverze** egymásnak, ha azonos bemeneti változó értékeihez tartozó függvény értékei egymás negáltjai.

Példa:

Duális párok:

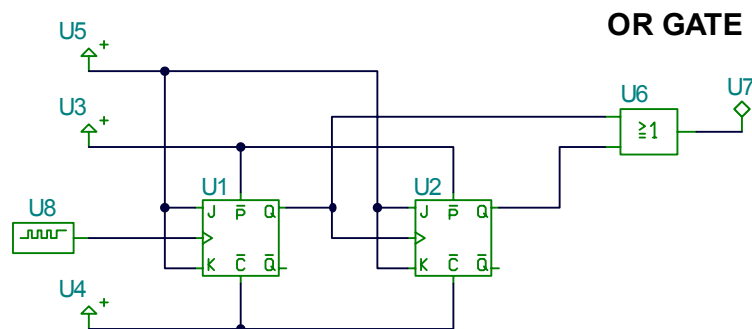
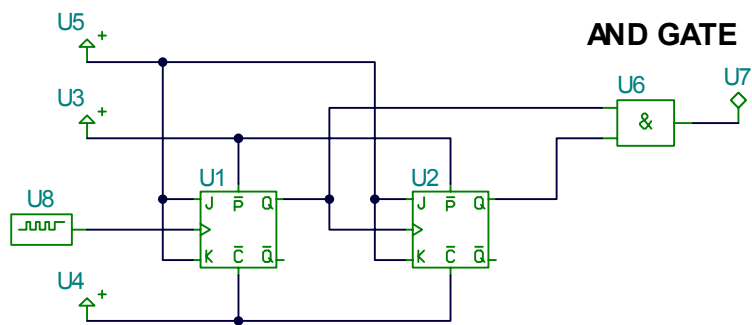
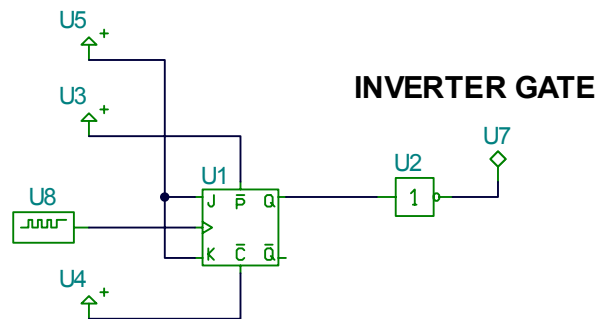
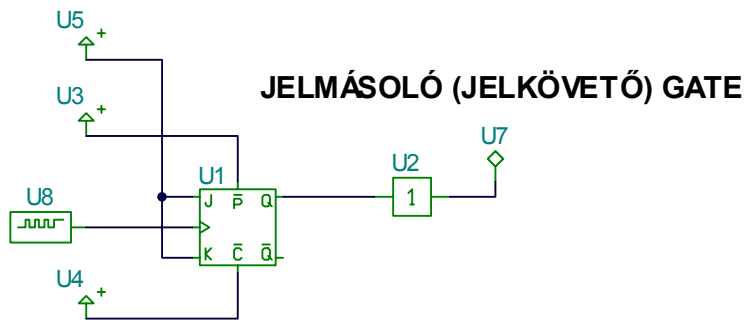
Inverz párok:

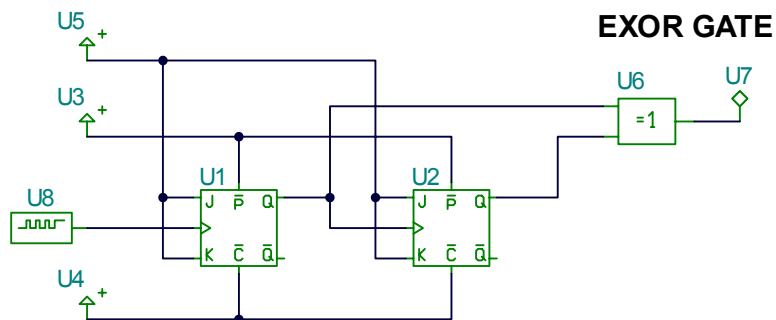
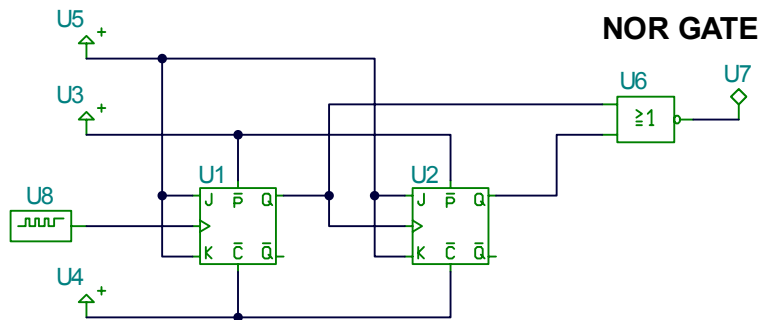
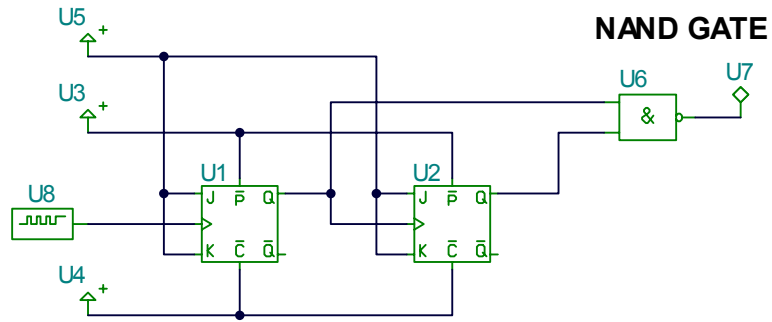
$$A \& B \rightarrow A+B$$

$$A \& B \rightarrow \overline{A \& B}$$

Az INHIBÍCIÓ és az IMPLIKÁCIÓ függvények felírhatók az AND, OR és az INVERZIÓ műveleteivel.

Az alábbi hálózatok, az egy-és minimálisan két logikai bemeneti változós alap kapuáramkörök működését szemléltetik.





Az **Igen; Nem; And; Or; Nand; Nor** és **Exor.sch** kiterjesztésű file-ok, működésben is nyomon követhetők.

A kapuáramkörök bemeneteit a vezérlési állapotoknak megfelelően, „1 ill. 2” bites JK F.F.-okból kialakított aszinkron számlálók vezérlik automatikusan, a működésük elsajátításának megkönnyítése céljából.

3. Több változós logikai függvények megadási módszerei

- kombinációs táblázat;
- KV tábla;
- teljes diszjunktív normál alak;
- mintermes megadási módszer;
- egyszerűsített mintermes megadási módon;
- ÉS/VAGY (AND/OR) hálózat MSZ jelképekkel;
- relés (vagy érintkezős) ÉS/VAGY (AND/OR) hálózat;
- NAND/NAND hálózat MSZ jelképekkel;
- teljes konjunktív normál alak;
- maxtermes megadási módszer;
- egyszerűsített maxtermes megadási módszer;
- VAGY/ÉS (OR/AND) MSZ jelképekkel hálózat;
- VAGY/ÉS (OR/AND) hálózat reléekkel;
- NOR/NOR hálózat MSZ jelképekkel;
- idődiagram;
- felcserélt bemeneti változójú megadási mód [$F^3_{212}(C,B,A)$].

Feladat: határozzuk meg az $F^3_{212}(A, B, C)$ függvényt a fenti megadási módok szerint!

$F^3_{212}(A,B,C)$

Kombinációs tábla

KV-tábla

212	0	$212_{(10)} = 11010100_{(2)}$
106	0	
53	1	
26	0	
13	1	
6	0	
3	1	
1	1	
0		

	2^2	2^1	2^0	
mi	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

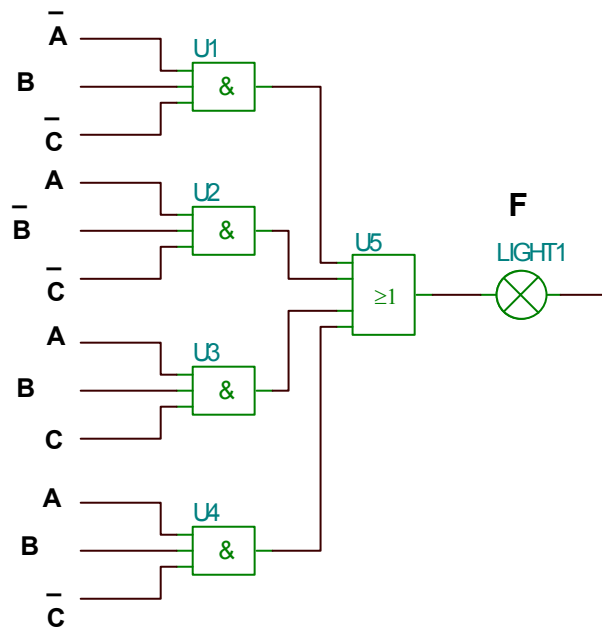
	$B_{(2)}$			
	0	1	3	1 ₂
$A_{(4)}$	1 ₄	5	1 ₇	1 ₆
	$C_{(1)}$			

Teljes diszjunktív normál alak, annyi elemi ÉS kapcsolat (minterm) VAGY kapcsolata ahány logikai „1”-est tartalmaz a függvény.

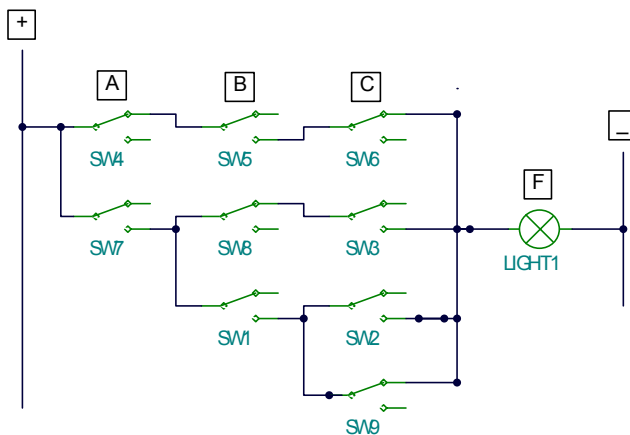
$$F^3_{212}(A,B) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC$$

A fenti függvény a $212_{(10)}$ szám teljes diszjunktív normál alakja.

AND/OR hálózat megvalósítása MSZ jelképekkel



AND/OR hálózat realizálása érintkezőkkel (relé kontaktusokkal)



Morzésítás szükséges és elégséges feltétele az, hogy a hálózat tartalmazza ugyanazon relé záró és bontó érintkezőjét, és legyen közösíthető pontjuk.

$$F_{212}^3(A,B,C) = m_2 + m_4 + m_6 + m_7 \quad \text{mintermes megadási módszer}$$

$$F_{212}^3(A,B,C) = \Sigma(2,4,6,7) \quad \text{egyszerűsített mintermes megadási módszer}$$

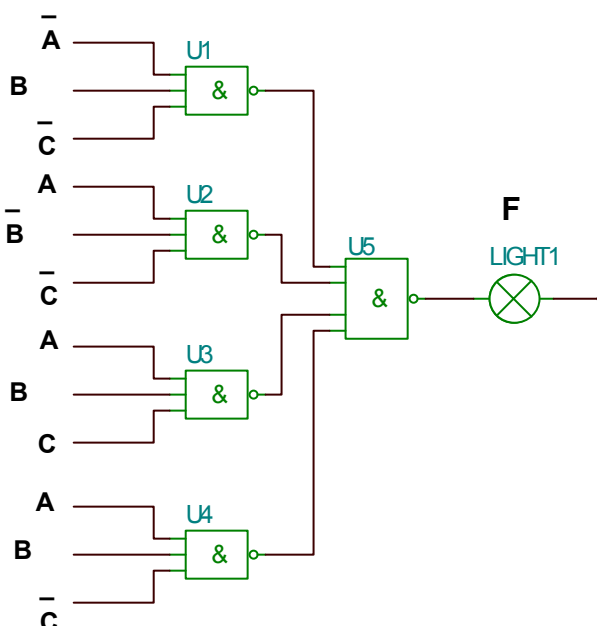
(általában a gyakorlat ezt használja)

NAND/NAND alakú függvény meghatározásához, a De Morgan szabály kétszeri alkalmazásával indulunk ki a diszjunktív alakú függvényből.

=

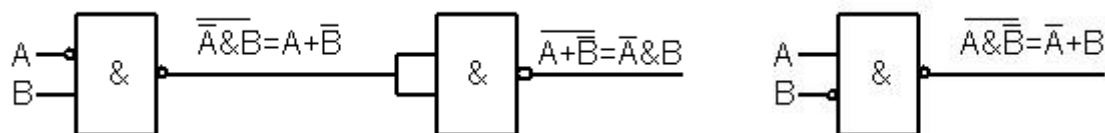
$$F_{212}^3(A, B, C) = \overline{\overline{\overline{ABC} + \overline{ABC} + ABC + \overline{ABC}}} = \overline{\overline{ABC} \& \overline{ABC} \& \overline{ABC} \& \overline{ABC}}$$

NAND/NAND hálózat realizálása MSZ jelképekkel



Példa: Realizáljuk NAND elemekkel a legegyszerűbb alakban az alábbi két függvényt

$$F = \overline{A} \cdot B; \quad F = \overline{A + B}$$



Teljes konjunktív normál alak, annyi elemi VAGY kapcsolat (maxterm) ÉS kapcsolata, ahány logikai „0”-át tartalmaz a függvény.

KV tábla

	B			
	0 ₀	0 ₁	0 ₃	2
A	4	0 ₅	7	6
	C			

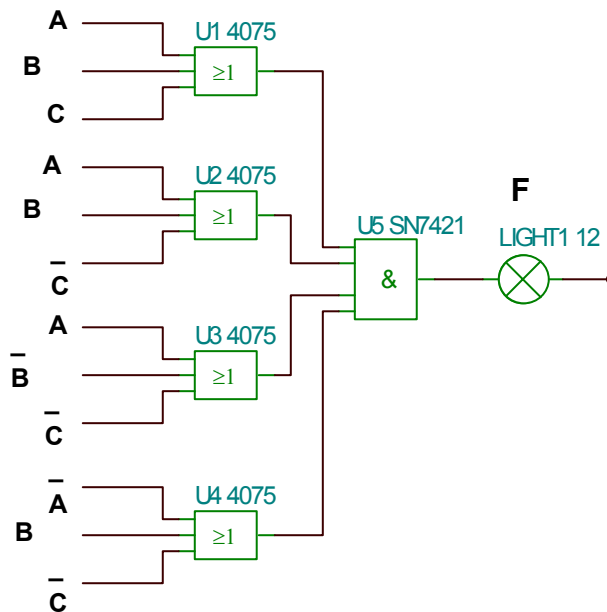
Teljes konjunktív normál alak:

$$F_{212}^3(A, B, C) = (A + B + C) \& (A + B + \overline{C}) \& (A + \overline{B} + \overline{C}) \& (\overline{A} + B + \overline{C})$$

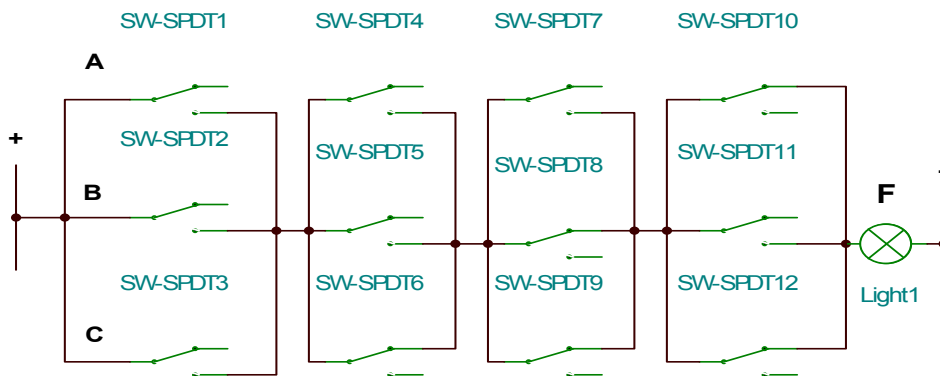
$$F_{212}^3(A, B, C) = \overline{\overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC}} = \\ = (A + B + C) \& (A + B + \overline{C}) \& (A + \overline{B} + \overline{C}) \& (\overline{A} + B + \overline{C})$$

A konjunktív alakú függvény felírása: az ($\overline{1} = 0$) Boole algebrai szabályból kiindulva az, hogy *amelyik változóban benne van abban nincs benne*, valamint *amelyik változóban nincs benne abban benne van*.

OR/AND hálózat megvalósítása MSZ jelképekkel



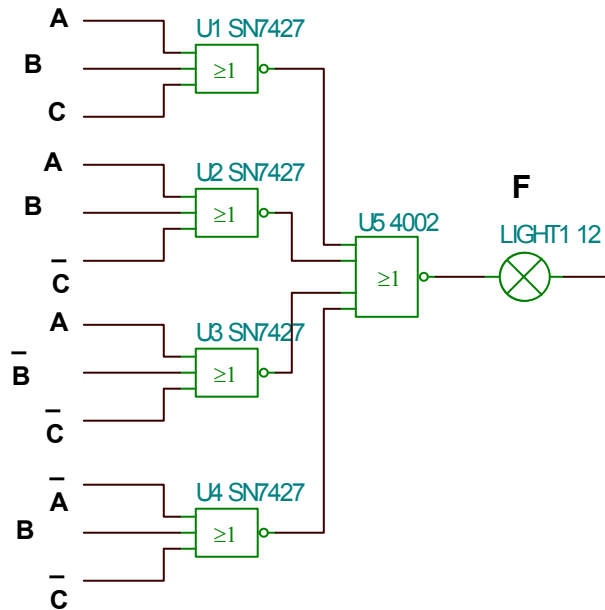
Érintkezős OR/AND hálózat



A NOR/NOR alakú függvény meghatározásához a konjunktív alakú függvényből indulunk ki a De Morgan szabály kétszeri alkalmazásával, melyet az alábbiak mutatják.

$$= \\ F_{212}^3 = \overline{\overline{(A + B + C) \& (A + B + \overline{C}) \& (A + \overline{B} + \overline{C}) \& (\overline{A} + B + \overline{C})}} = \\ = \overline{\overline{(A + B + C)} + \overline{\overline{(A + B + \overline{C})}} + \overline{\overline{(A + \overline{B} + \overline{C})}} + \overline{\overline{(\overline{A} + B + \overline{C})}}}$$

A NOR/NOR hálózat megvalósítását MSZ jelképekkel az alábbi hálózat szimbolizálja



Maxtermes megadási módszer

A keresett maxterm szám úgy határozható meg, hogy a maximális m_i -számból kivonjuk a keresett maxtermszámot, majd a keresett maxterm, abban a m_i -ben található (pl M_2 három változó esetében az $m_7 - M_2 = m_5$, tehát az M_2 -es maxterm az m_5 -ös mintermben található).

	B			
	7	6	4	0
	$\bar{0}_0$	$\bar{0}_1$	$\bar{0}_3$	2
A	3	2	0	1
	4	$\bar{0}_5$	7	6
	C			

$$F_{212}^3(A,B,C) = M_2 \& M_4 \& M_6 \& M_7$$

↑
Maxtermes megadási módszer

Egyszerűsített maxtermes megadási módszer (a gyakorlatban általában ez használatos)

$$F_{212}^3(A,B,C) = \Pi(2,4,6,7)$$

Három változós (A,B,C) logikai függvények mintermjei és maxtermjei közötti összefüggések

$F(A, B, C)$

$$m_0 = \bar{A}\bar{B}\bar{C}$$

$$M_0 = \bar{A} + \bar{B} + \bar{C}$$

$$m_1 = \bar{A}\bar{B}C$$

$$M_1 = \bar{A} + \bar{B} + C$$

⋮

⋮

⋮

⋮

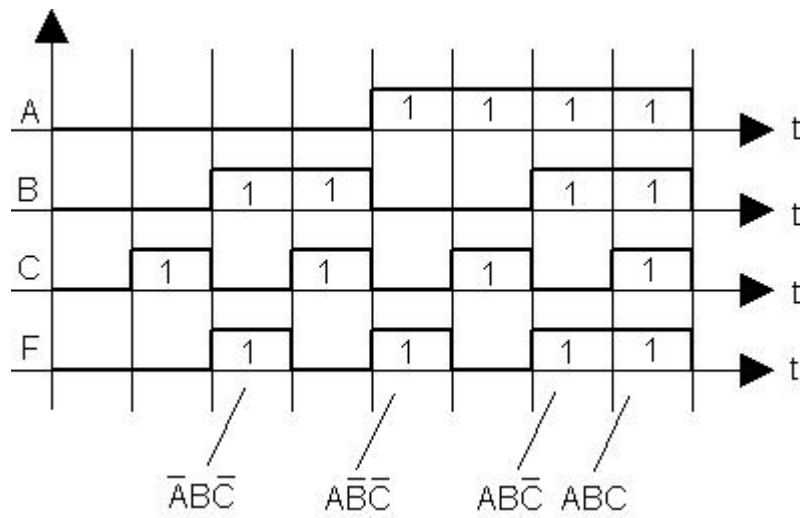
$$m_7 = ABC$$

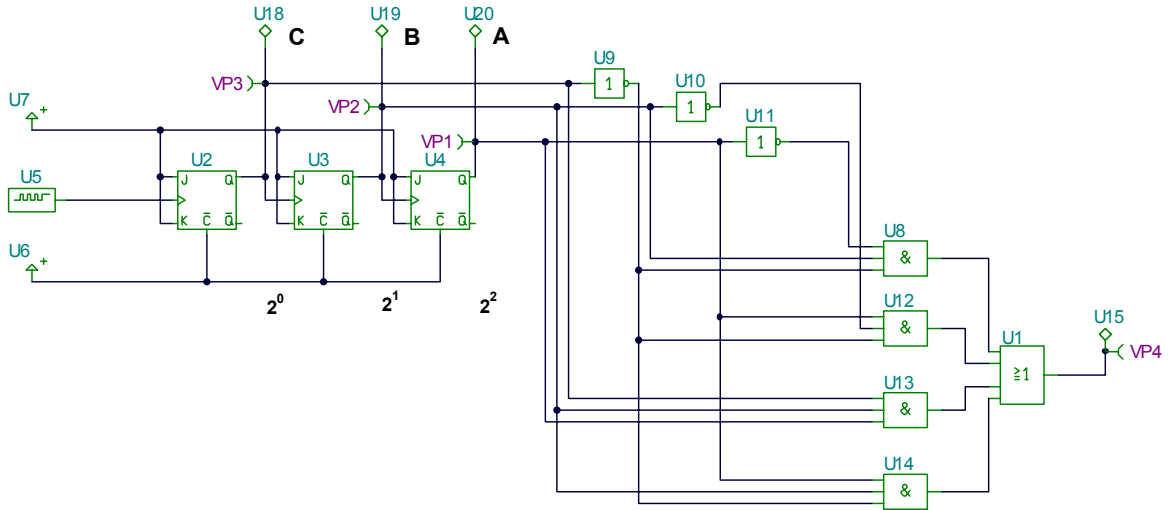
$$M_7 = A + B + C$$

Indexszámos megadási módszer

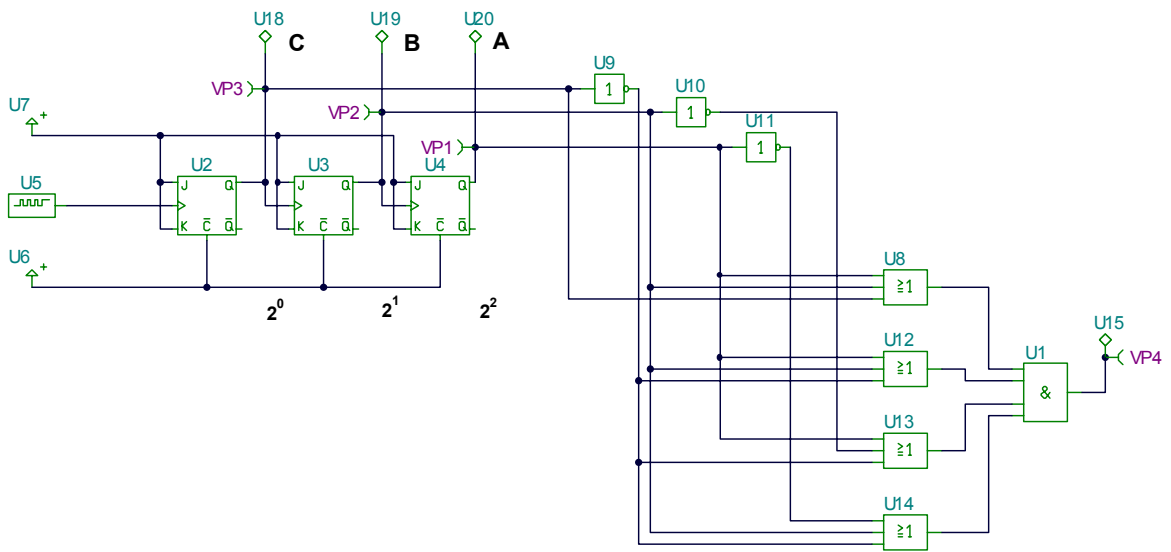
$$F_{212}^3(A,B,C) = 1 \cdot 2^2 + 1 \cdot 2^4 + 1 \cdot 2^6 + 1 \cdot 2^7 = 212_{(10)}$$

Idődiagram



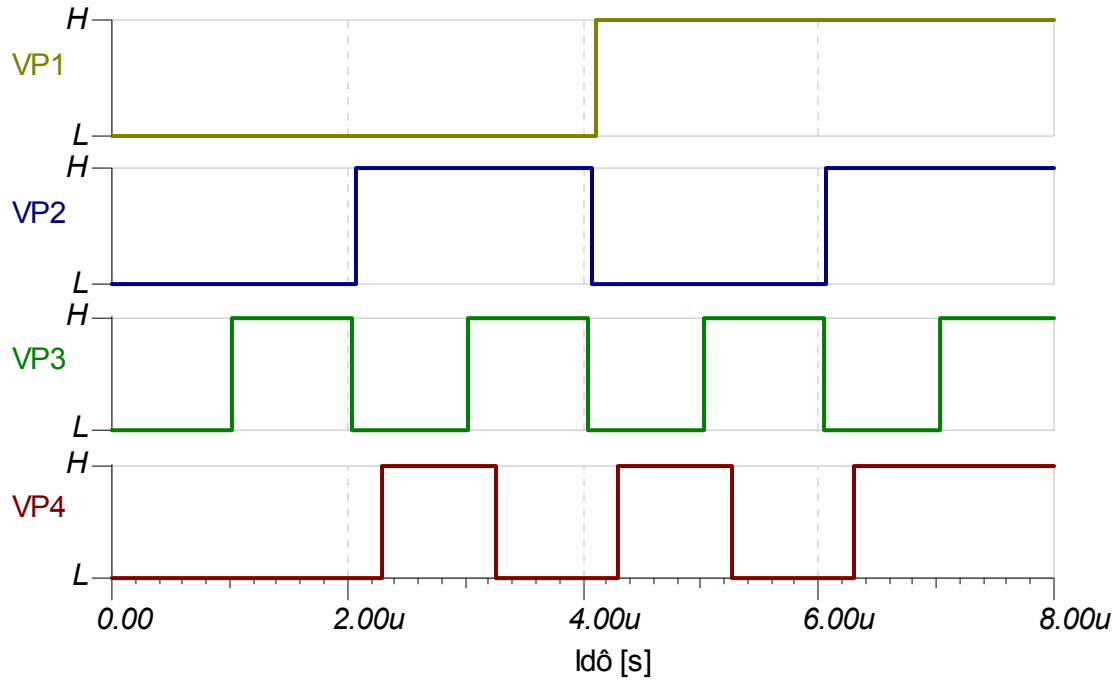


A függvény teljes diszjunktív normál alakjának AND/OR realizációja



A függvény teljes konjunktív normál alakjának OR/AND realizációja

Az AND/OR és az OR/AND hálózatok működés közben is szimulálják, az F^3_{212} (A, B, C) logikai függvényt.



Az F^3_{212} (A, B, C) logikai függvény tényleges idődiagramja.

Kérdés: van-e, vagy vannak-e olyan függvények, ahol a m_i számok megegyeznek a M_i számokkal? Ha igen (példa) és ha nem miért?

Példa: a fenti kérdés magyarázatára, válasz végtelen sok ilyen logikai függvény van

F	m_i	M_i
0	0	7
1	1	6
0	2	5
1	3	4
0	4	3
1	5	2
0	6	1
1	7	0

$$\Sigma (1,3,5,7)$$

$$\Pi (1,3,5,7)$$

Példa:

$$F_{186}^3(A,B,C) = 1 \cdot 2^1 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^7$$

	mi	A	B	C	F
186	0	0	0	0	0
93	1	0	0	1	1
46	2	0	1	0	0
23	3	0	1	1	1
11	4	1	0	0	1
5	5	1	0	1	1
2	6	1	1	0	0
1	7	1	1	1	1

		B			
		0	1 ₁	1 ₃	2
A		1 ₄	1 ₅	1 ₇	6
		C			

Teljes konjunktív normál alak

$$F_{186}^3(A,B,C) = (A + B + C) \& A + \bar{B} + C) \& (\bar{A} + \bar{B} + C)$$

4. Logikai függvények egyszerűsítése, minimalizálása

Lehetséges alternatívák:

- algebrai út (Boole algebrai azonosságok alapján);
- grafikus út (KV) táblán maximum hat változóig;
- numerikus módszer (Quine & Mc Cluskey).

Algebrai egyszerűsítés:

$$F^2(A, B) = \overline{A}\overline{B} + \overline{A}B + A\overline{B} + AB$$

$$= \overline{A} \& \underbrace{(\overline{B} + B)}_1 + A \& \underbrace{(\overline{B} + B)}_1 = \overline{A} + A = 1$$

$$F^2(A, B) = A \& (A + B) = AA + AB = A \cdot \underbrace{(1 + B)}_1 = A$$

$$A + B + A \oplus B = A + B + AB + \overline{A}\overline{B} = A(1 + B) + B(1 + \overline{A}) = A + B$$

Az egyszerűsítés célja:

- minimalizált kapuszám (IC tok)
- kevesebb relé érintkező
- kisebb a meghibásodás lehetősége
- könnyebb javíthatóság
- olcsóbb a kivitelezés

Karnaugh-Veitch (KV)

Grafikus egyszerűsítés diszjunktív alakban: a függvény 1-el jelölt helyeiből indulunk ki.

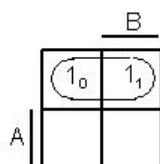
konjunktív alakban: a függvény 0-val jelölt helyeiből indulunk ki

Lehetséges tömbösítések, lefedések.

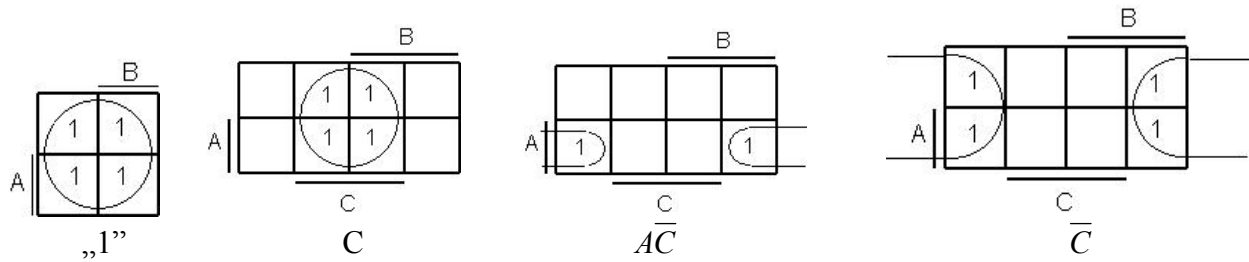
A KV- tábla a szimmetria tengelyére, ha képzeletben összehajtanánk, akkor a szomszédos mintermek („1”), vagy maxtermek („0”) lefedhetők egy közös tömbbel.

Mindig a lehető legnagyobb tömböket alakítjuk ki, a legegyszerűbb alak kialakításának érdekében, valamint ha nem teljesen specifikált a logikai függvény, azaz, ha x-eket is tartalmaz, akkor a lehető legnagyobb tömbök kialakításához, ezeket is bevonjuk a tömbösítésbe. Természetesen külön x-eket nem tömbösítünk, mivel ezek közömbösek a bemeneti változók lehetséges kombinációitól.

Példák: lehetséges lefedésekre két és három változó esetében:



$$\overline{A}\overline{B} + \overline{A}B = \overline{A} \& \underbrace{(\overline{B} + B)}_1 = \overline{A}$$



Redundáns elem: (közömbös) a kimenet független a bemeneti változók kombinációjától.

Jelölése: x. Valamely üzenetforrás ki nem használt információ tartalma.

Azok a függvények, amelyek x-et is tartalmaznak nem teljesen specifikáltak (nem teljesen határozottak).

Grafikus egyszerűsítés lépései:

1. KV táblán ábrázoljuk a függvényt
(ha x elemet is tartalmaz, azt is ábrázoljuk x-ként)
redundáns

minden szomszédos $\left. \begin{array}{l} \text{1-el jelölt } m_i\text{-ket} \\ \text{0-val jelölt } M_i\text{-ket} \end{array} \right\}$ közös tömbbé kell összevonnunk

mindig a lehető legnagyobb tömböket kell elkészítenünk a legegyszerűbb alak létrehozása érdekében. Tömbösítésbe az x-et is belevonjuk a lehető legnagyobb tömbök elkészítéséhez, a legegyszerűbb alak létrehozása érdekében.

Külön x-eket nem tömbösítünk!

(„1” m_i -n
„0” M_i -n)

Minden „1”-el, vagy „0”-val jelölt cellát (mintermet, maxtermet) legalább 1-szer 1 hurokkal le kell fedni. Azok a cellák, amin csak 1 hurok megy keresztül, jelöljük meg egy ponttal és ezt nélkülözhetetlen primimplikásnak nevezzük ami a függvény megvalósításához elengedhetetlen..

A négyváltozós KV tábla mintermjének az elhelyezkedése:

C_2

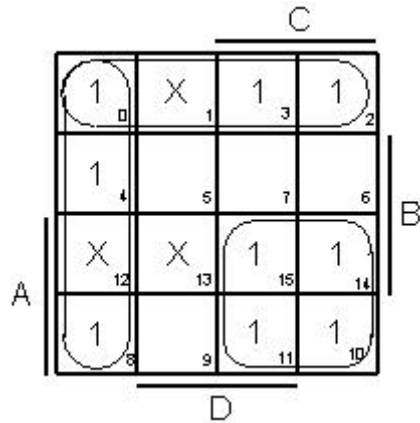
	0	1	3	2	
	4	5	7	6	
	12	13	15	14	B_4
A_8	8	9	11	10	
					D_1

Feladat: minimalizáljuk az alábbi függvényt diszjunktív és konjunktív alakban!

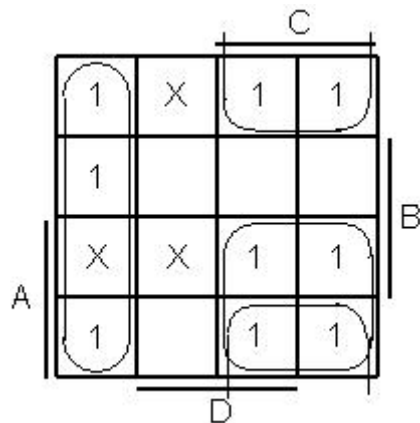
$$F(A,B,C,D)=\Sigma(0,2,3,4,8,10,11,14,15)+\Sigma_x(1,12,13)$$

A diszjunktív alakban egyszerűsített függvényeket realizáljuk ÉS/VAGY (MSZ jelképekkel), érintkezős és NAND/NAND hálózatokkal, a konjunktív alakban egyszerűsített függvényeket pedig: VAGY/ÉS, érintkezős és NOR/NOR hálózatokkal.

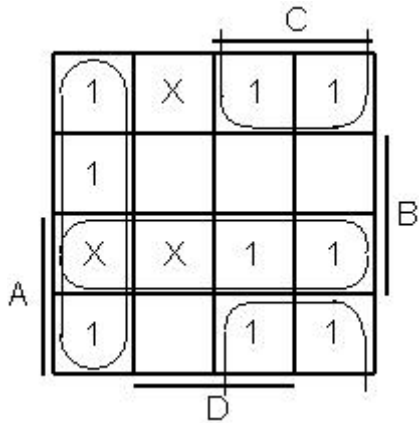
Lehetséges tömbösítések:



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

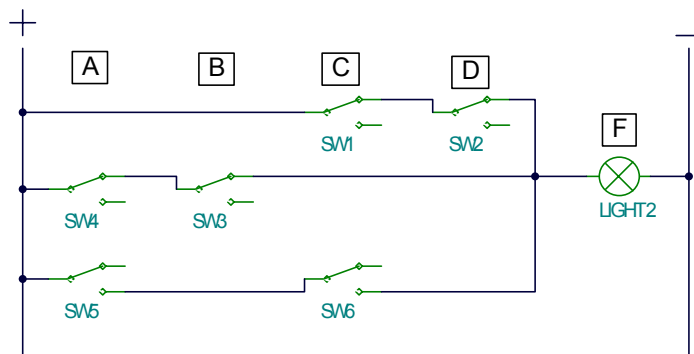


$$F(A,B,C,D)=\overline{C}\overline{D} + AC + \overline{B}C$$

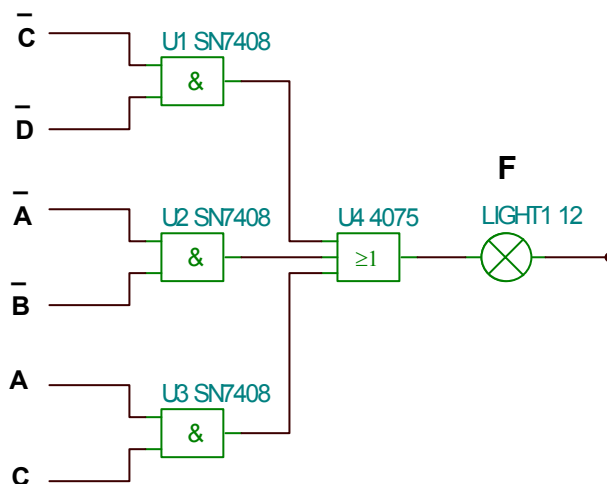


$$F(A,B,C,D) = \overline{C}\overline{D} + AB + \overline{B}C$$

Érintkezős megvalósítás



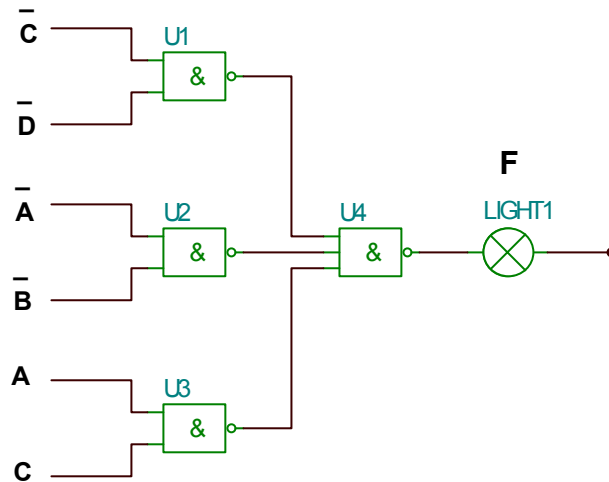
ÉS/VAGY megvalósítás



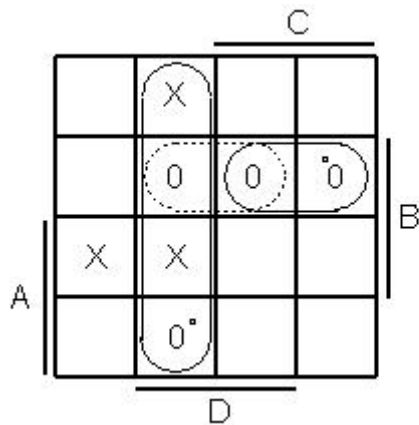
NAND/NAND alakú függvény meghatározásához, a De Morgan szabály kétszeri alkalmazásával juthatunk el, az alábbiak szerint.


$$\overline{\overline{F(A,B,C,D)}} = \overline{\overline{CD + \overline{AB} + AC}} = \overline{\overline{CD} \& \overline{AB} \& \overline{AC}}$$

NAND/NAND hálózat kialakítása MSZ jelképekkel



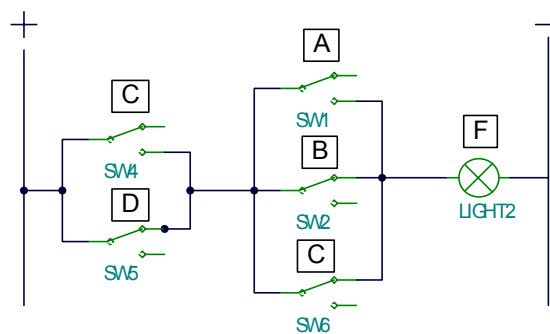
Konjunktív alakban történő egyszerűsítés menete az, hogy a függvény „0” helyeiből indulunk ki.



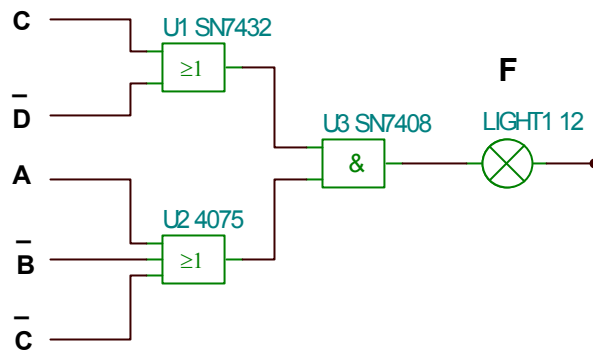
 -nem nélkülözhetetlen primimplikáns

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

Érintkezős megvalósítás



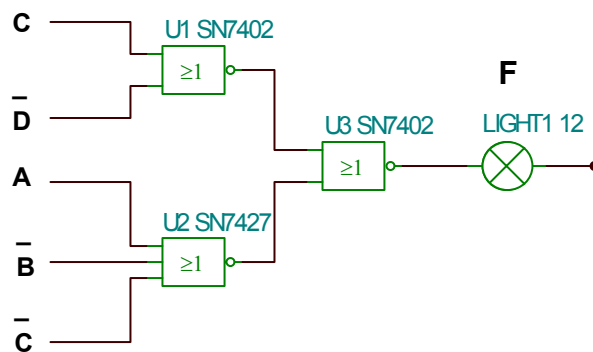
VAGY/ÉS realizáció



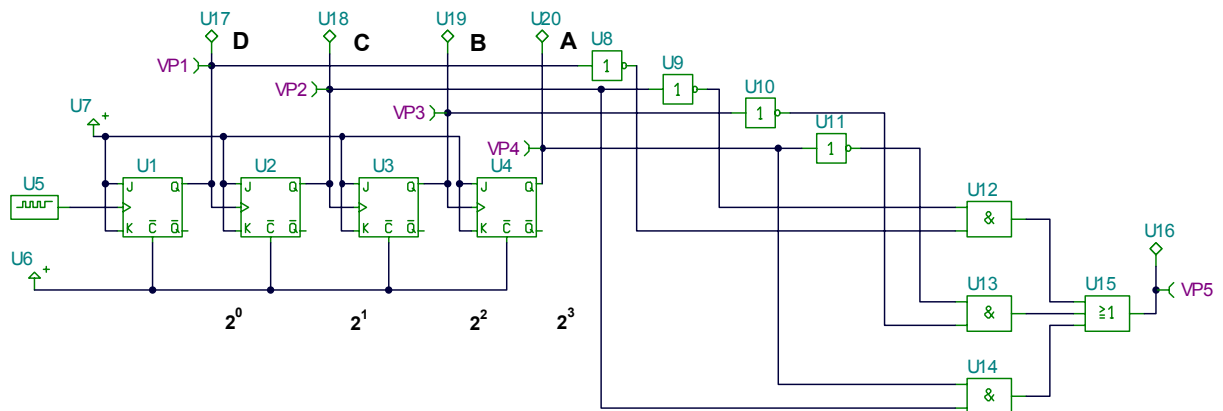
A NOR/NOR alakú függvény meghatározásához a De Morgan szabály kétszeri alkalmazásával jutunk el, az alábbi módon.

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

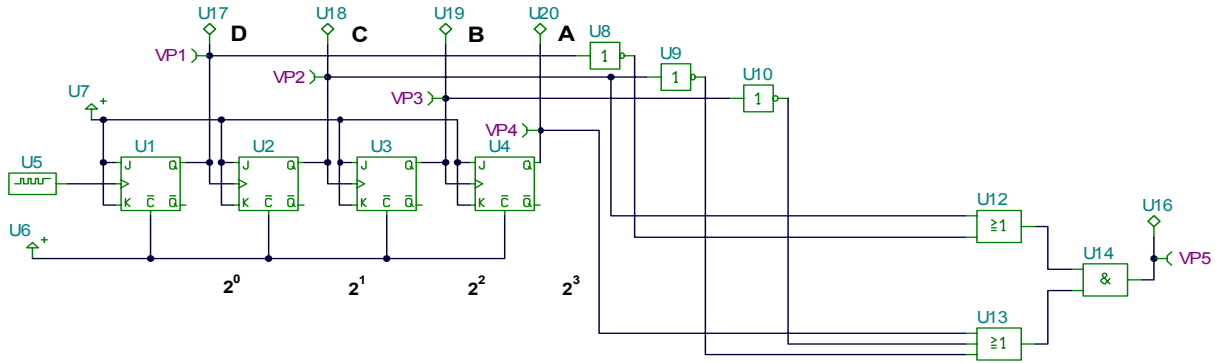
NOR/NOR hálózat kialakítása MSZ szimbólumokkal



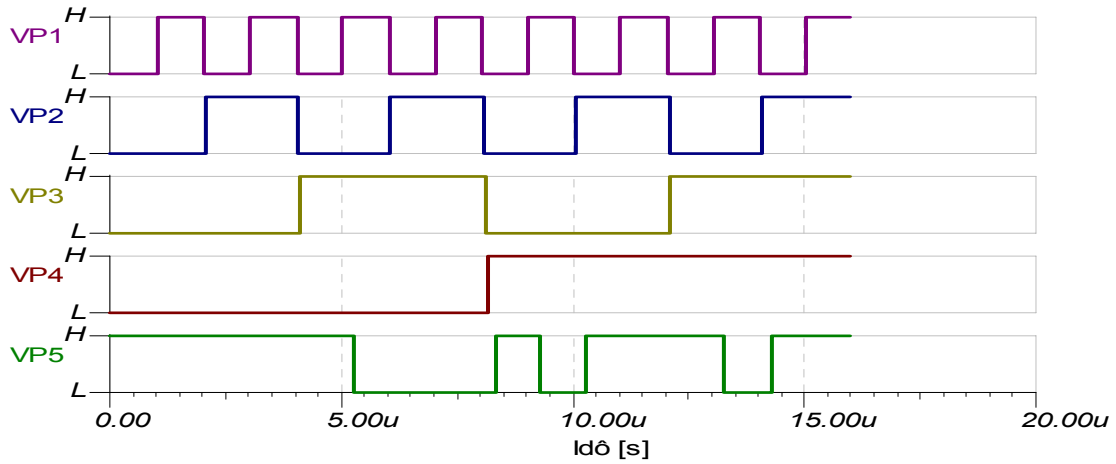
A lenti két hálózat, az $F(A, B, C, D) = \sum (0, 2, 3, 4, 8, 10, 11, 14, 15) + \sum_x (1, 12, 13)$ függvény AND/OR és OR/AND realizációit tartalmazza, a működést szimuláló idődiagrammal



A diszjunktív alakban egyszerűsített függvény AND/OR realizációja

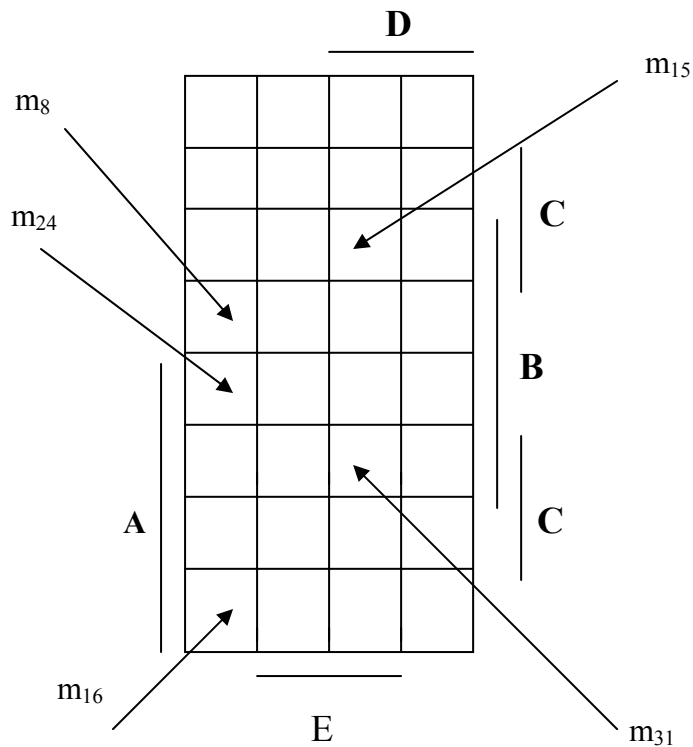


Akonjunktív alakban egyszerűsített függvény OR/AND realizációja



Az alábbi síkbeli KV-táblák az öt, ill. hat változós függvények elrendezést szemléltetik.

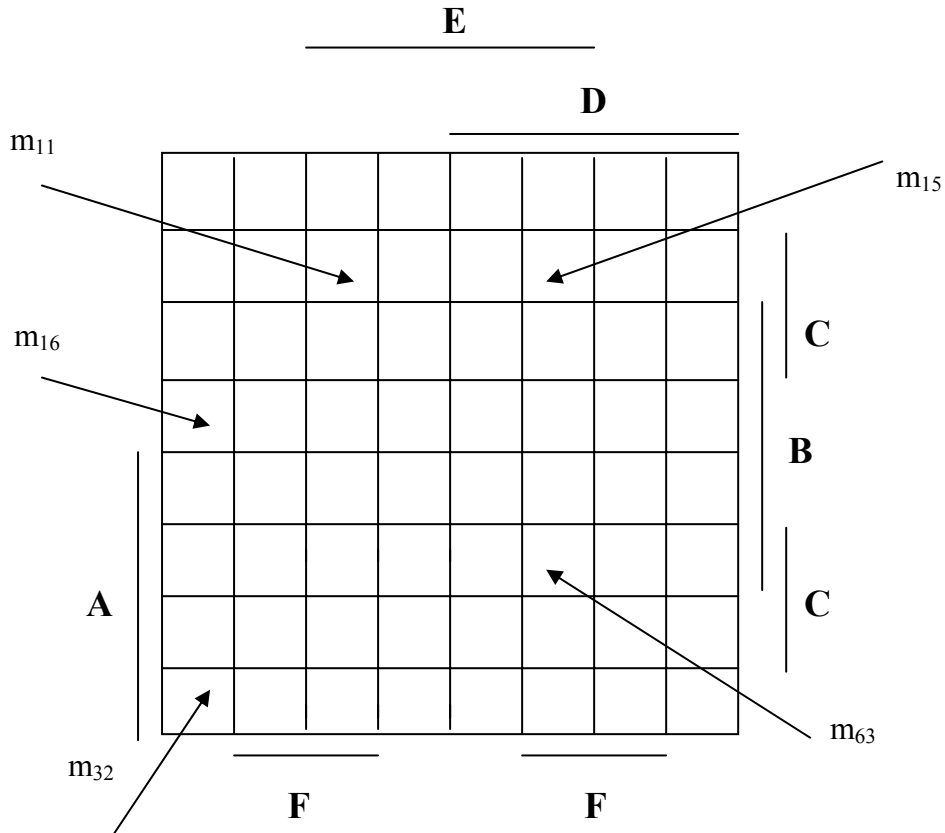
Öt változós F (A, B, C, D, E) logikai függvény síkbeli KV táblájának elrendezése, peremezése.



Súlyozása: 16, 8, 4, 2, 1

A B C D E

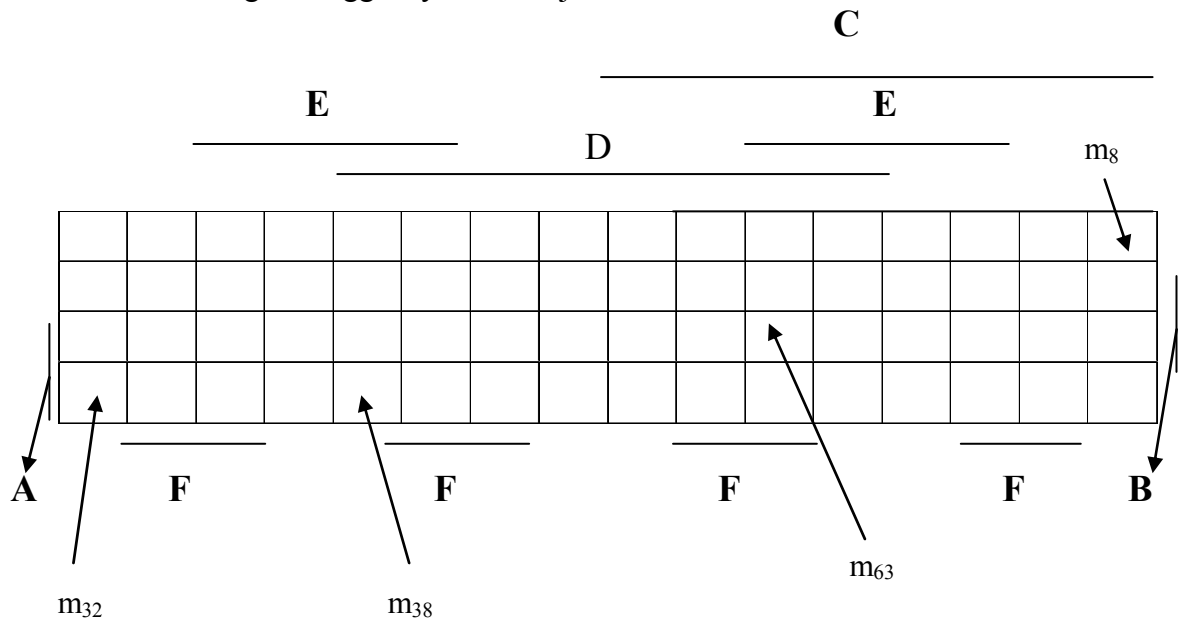
Hat változós F (A, B, C, D, E, F) logikai függvény síkbeli KV táblájának elrendezése, peremezése.



Súlyozása: 32, 16, 8, 4, 2, 1

A B C D E F

Szintén hat változós logikai függvény KV táblájának az elrendezését szemlélteti az alábbi ábra.



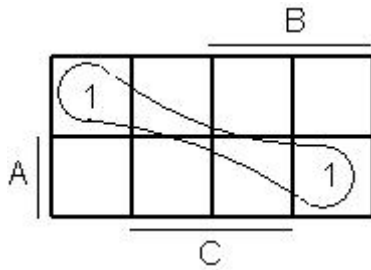
Súlyozása: 32, 16, 8, 4, 2, 1

A B C D E F

Megjegyzés a grafikus eljárás maximálisan 6 változóiig használható.

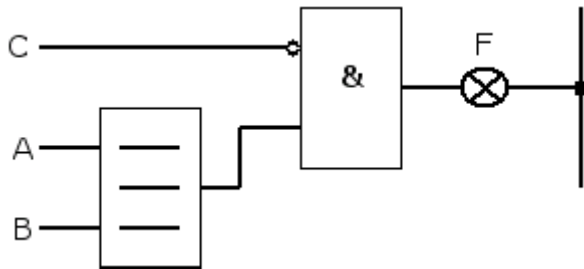
EKVIVALENCIA egyszerűsítési módszer

$$F^3(A,B,C) = \Sigma(0,6)$$

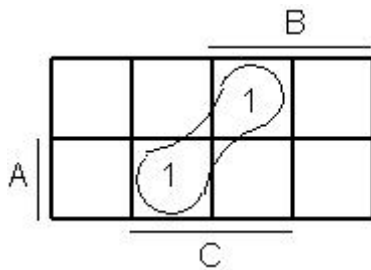


$$F^3(A,B,C) = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} = \bar{C} \& (A\bar{B} + \bar{A}B)$$

Megvalósítás MSZ jelképekkel

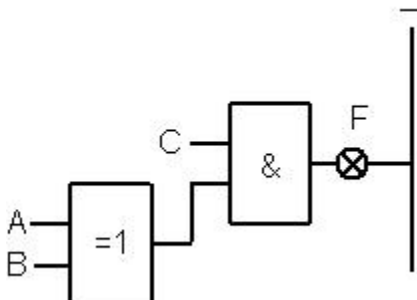
**ANTIVALENCIA egyszerűsítési módszer**

$$F^3(A,B,C) = \Sigma(3,5)$$



$$F^3(A,B,C) = \bar{A}BC + A\bar{B}C = C \& (\bar{A}B + A\bar{B}) = C \& (A \oplus B)$$

Megvalósítás MSZ jelképekkel



5. Kódolási alapfogalmak

- kód: két különböző szimbólumhalmaz egyértelmű egymáshoz rendelése
- szimbólumhalmaz: bináris kód esetén 0,1
- kódszó: szimbólumok meghatározott sorozata
- kódolást: a kódoló áramkör (Encoder) végzi
- dekódolást a dekódoló áramkör (Decoder) végzi
- az információ mennyiség (H) egysége bit/szimbólum (más elnevezésben: a rendszer entrópiája)

Valószínűség: (p): kedvező esetek száma/összes esetek száma: $p=k/n$

$$H_{\max} = \sum_{i=1}^n p_i \log_2 \frac{1}{p_i} \quad \text{Shannon törvénye; mértékegysége: bit/szimbólum}$$

- redundancia x : valamely üzenetforrás ki nem használt információ tartalma
- a rendszer max. entrópiája

$$R = H_{\max} - H$$

- relatív redundancia

$$R_{\text{rel}} = \frac{H_{\max} - H}{H_{\max}} \cdot 100 \%$$

- Hamming távolság: két kódszó között annyi, amennyivel meg kell változtatni a másodikat, hogy az előzőt kapjuk
pl.

101
100
H=1

- paritásbit (ellenőrző bit): ha a kódszóban az 1-ek száma páros a paritás 0
ha a kódszóban az 1-ek száma páratlan a paritás 1

Kódszó	P
1100	0
1101	1

Az információ átvitele átviteli csatornán keresztül történik

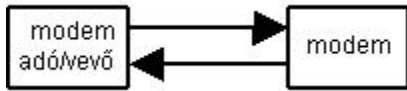


ADÓ: információt generáló egység

VEVŐ: információt fogadó egység



kétirányú adatátvitel esetén : modem



AM: amplitúdó modulált jel
FM: frekvencia modulált jel

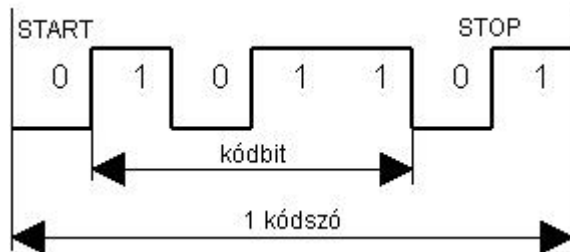
- az átviteli csatorna lehet:
 - soros:** a kódszavak egymást követően kerülnek átvitelre
 - párhuzamos:** a kódszavak egyidejűleg kerülnek átvitelre

Az adatátvitel lehet:

- szinkron üzemű: egyidejűleg történik meg a kódszó továbbítása blokkokban;
- aszinkron üzemű: **START** és **STOP** bitekkel egészül ki a kódszó és az átvitel egymást követően történik.

STOP bit: általában 2 bit hosszúságú.

START bit: 1 bit hosszúságú.



6. Alapkódok bemutatása és a képzési szabályuk

A **BINÁRIS** kód 2 hatványai szerint képezhető, az alábbi módon:

Példa:

$65_{(10)}$ szám felírása bináris számrendszerben

64	32	16	8	4	2	1
1	0	0	0	0	0	1

- **NBCD**
- **STIBITZ**
- **AIKEN**
- **GRAY**
- **JOHNSON**
- **Hexadecimális**

NBCD kód: normál BCD kód képzési szabálya

0-9-ig 10 db kódszó készlete van

I	mi	A	B	C	D
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1

$$142_{(10)} = 1\ 0100\ 0010_{(NBCD)}$$

STIBITZ kód képzési szabálya

Képzési szabálya: N+3 bináris; 10 db kódszó készlete van, 0-9-ig

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

$$142_{(10)} = 100\ 0111\ 0101_{(\text{STIBITZ})}$$

AIKEN kód képzési szabálya:

N=4-ig bináris
 N=5-9-ig N+6 bináris
 10 kódszóból áll 0-9

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

$$145_{(10)} = 1\ 0100\ 1011_{(\text{AIKEN})}$$

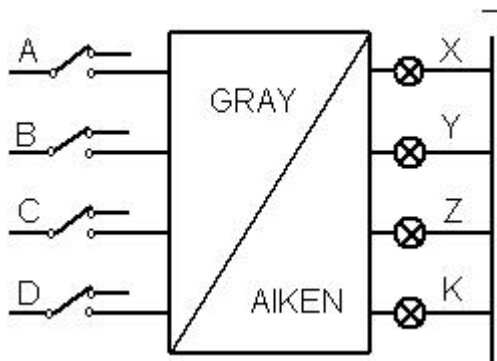
JOHNSON kód képzése 4biten

I	mi	A	B	C	D
0	0	0	0	0	0
1	1	0	0	0	1
2	3	0	0	1	1
3	7	0	1	1	1
4	15	1	1	1	1
5	14	1	1	1	0
6	12	1	1	0	0
7	8	1	0	0	0

Képzése 5 biten

A	B	C	D	E
0	0	0	0	0
0	0	0	0	1
0	0	0	1	1
0	0	1	1	1
0	1	1	1	1
1	1	1	1	1
1	1	1	1	0
1	1	1	0	0
1	1	0	0	0
1	0	0	0	0

Feladat: tervezzük meg azt a kombinációs hálózatot (kódátalakítót), amit az alábbi séma szemléltet.



A kódátalakító tervezésének menete:

- kombinációs táblázat felvétele;
- logikai függvények diszjunktív alakban való egyszerűsítése;
- az egyszerűsített függvények realizálása NAND/NAND hálózattal;
- az egyszerűsített függvények realizálása érintkezős hálózattal.

A kombinációs hálózat a kódátalakítást 0-8-ig valósítsa meg!

Kombinációs táblázat felvétele

	8	4	2	1					
m_i	I	A	B	C	D	X	Y	Z	K
0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	1
3	2	0	0	1	1	0	0	1	0
2	3	0	0	1	0	0	0	1	1
6	4	0	1	1	0	0	1	0	0
7	5	0	1	1	1	1	0	1	1
5	6	0	1	0	1	1	1	0	0
4	7	0	1	0	0	1	1	0	1
12	8	1	1	0	0	1	1	1	0

Redundáns m_i -k: 8, 9, 10, 11, 13, 14, 15

A függvények leolvasása, a m_i számok szerint történik

A kódátalakító diszjunktív alakban egyszerűsített logikai függvényeinek NAND/NAND alakja:

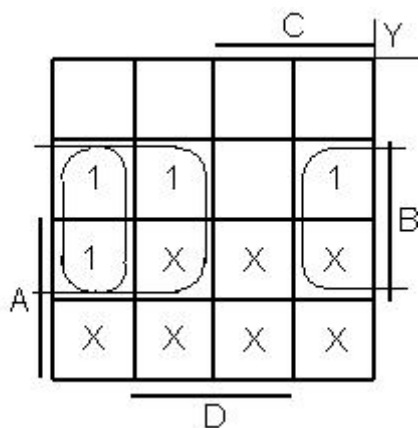
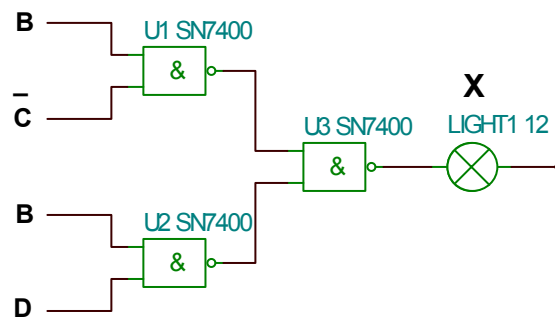
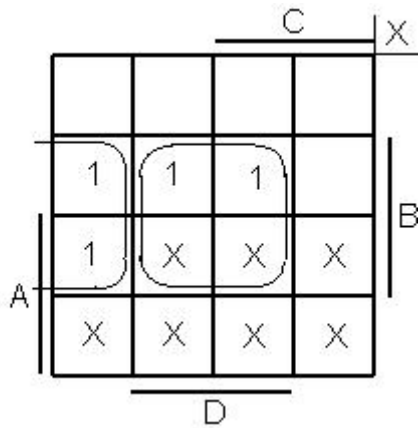
$$\overline{X} = \overline{\overline{CB} + \overline{BD}} = \overline{\overline{CB} \& \overline{BD}}$$

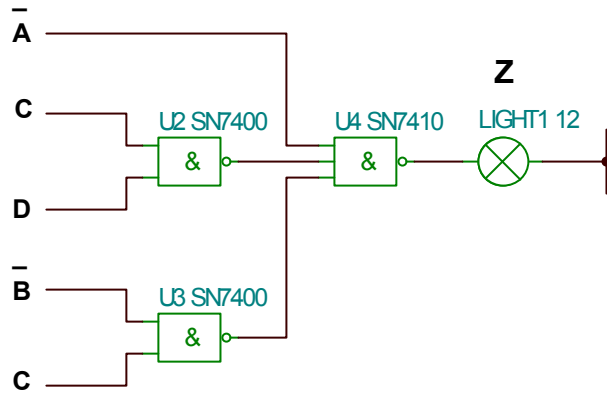
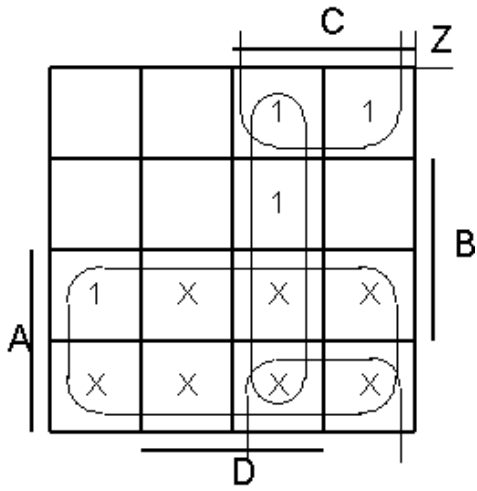
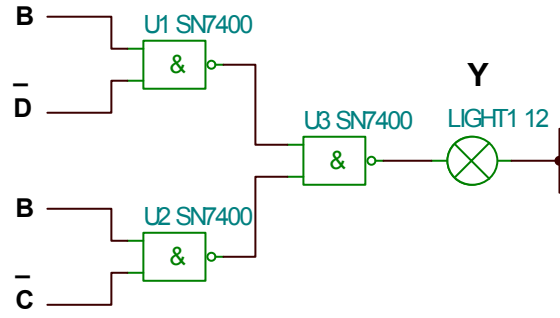
$$\overline{Y} = \overline{\overline{BD} + \overline{CB}} = \overline{\overline{BD} \& \overline{CB}}$$

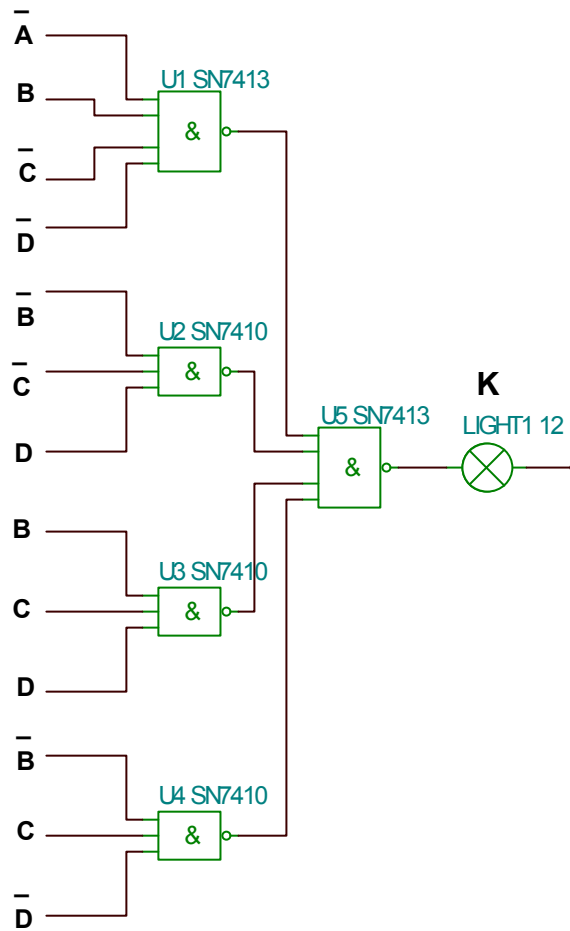
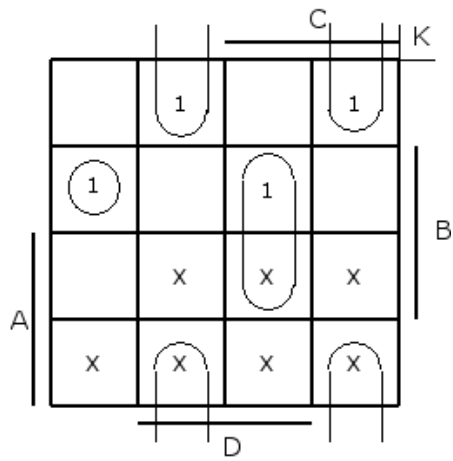
$$\overline{\overline{Z}} = \overline{\overline{A + CD + \overline{BC}}} = \overline{\overline{A} \& \overline{CD} \& \overline{\overline{BC}}}$$

$$\overline{\overline{K}} = \overline{\overline{\overline{ABC\overline{D}} + \overline{B\overline{C}D} + B\overline{C}D + \overline{BC\overline{D}}}} = \overline{\overline{\overline{ABC\overline{D}} \& \overline{\overline{B\overline{C}D}} \& \overline{\overline{B\overline{C}D}} \& \overline{\overline{BC\overline{D}}}}}$$

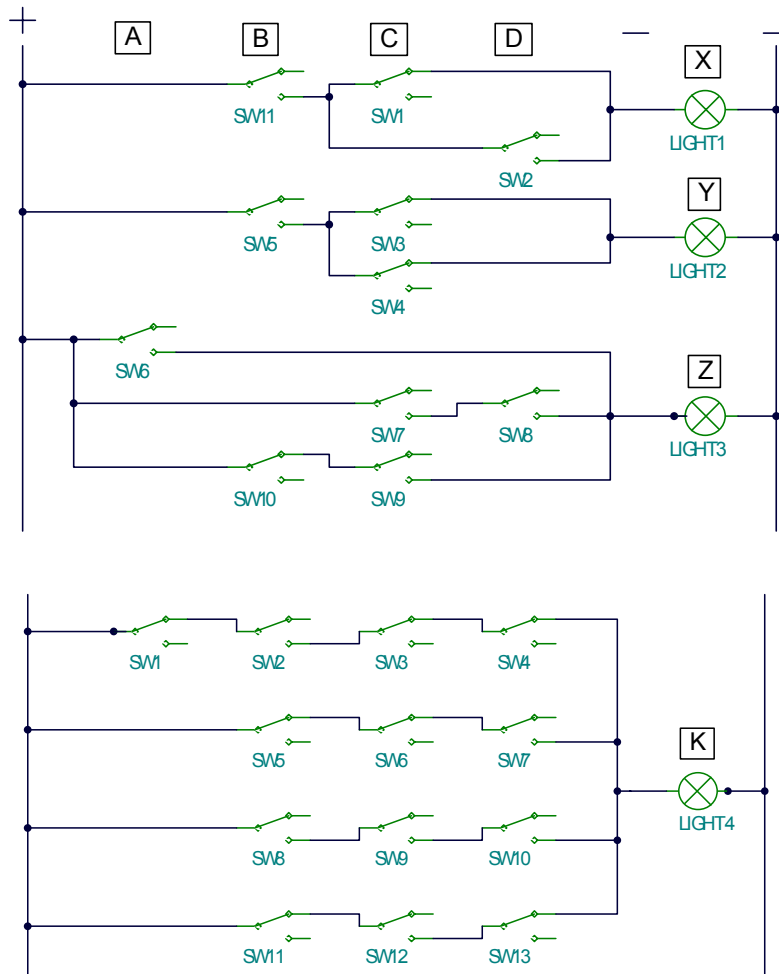
A diszjunktív alakban egyszerűsített kimeneti logikai függvények (X,Y,Z,K) KV táblái és az MSZ jelképekkel történő NAND/NAND realizálásuk látható az alábbiakban:

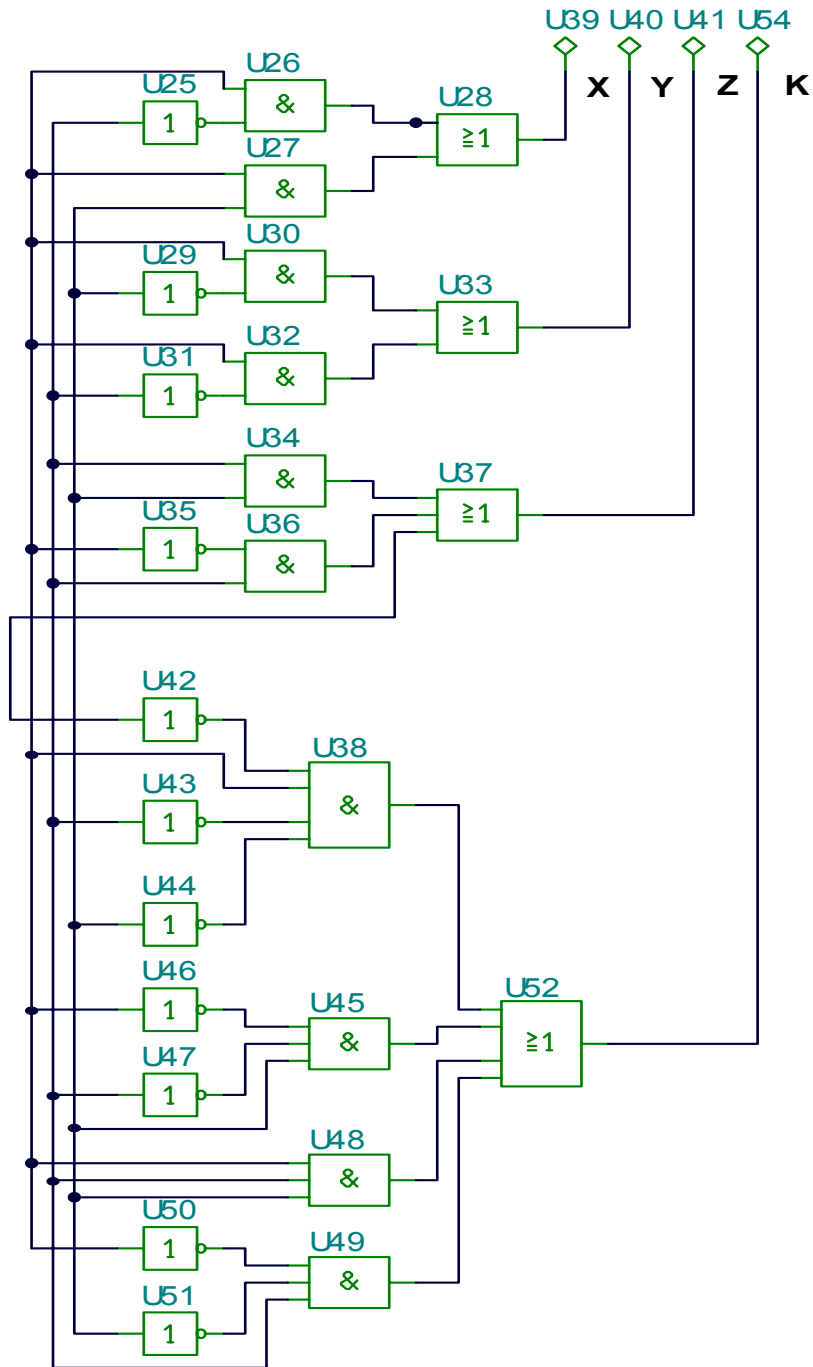




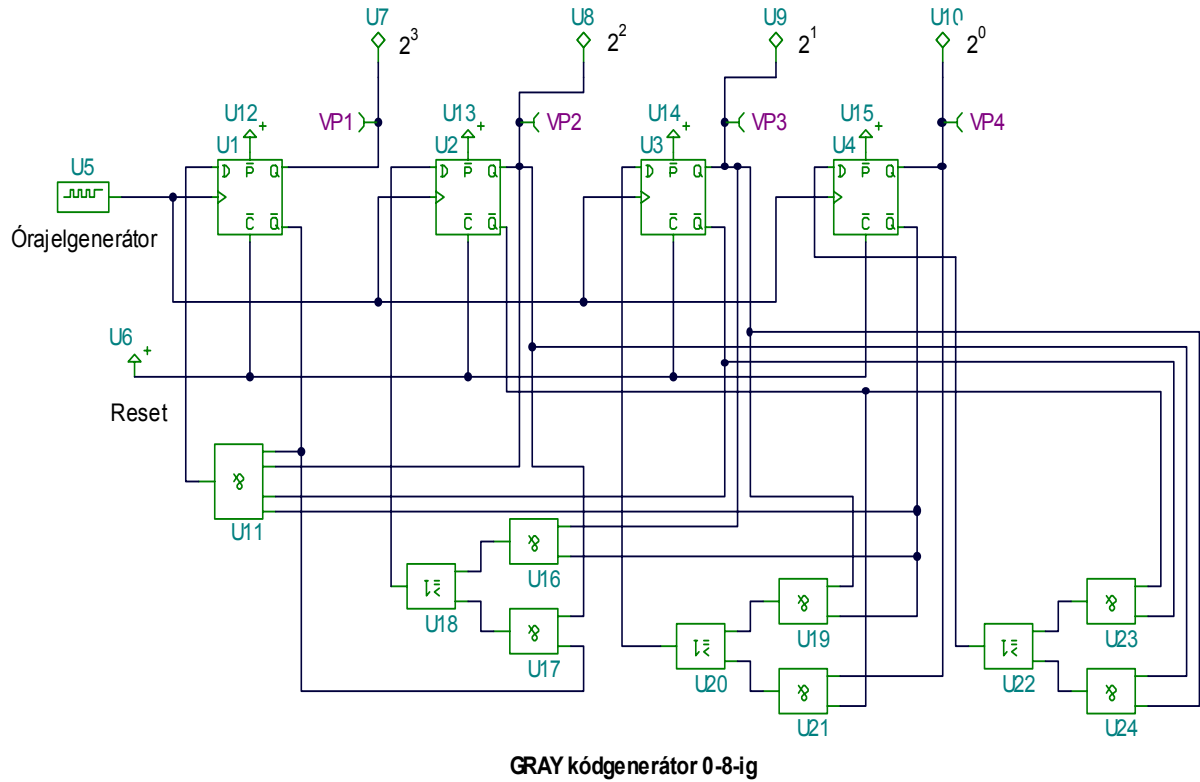


Érintkezős realizáció:





A fenti hálózat szimulációjával, a GRAY/AIKEN kódátalakító 0-8-ig, működés közben is nyomon követhető.



A fenti hálózat a bemeneti GRAY kódot generálja 0-8-ig, ami a kódátalakító automatizált működését teszi lehetővé.

7. Digitális áramkörök

Logikai egységek áramköri jellemzői

Az alábbiakban a logikai egységek áramköri leírásánál használt fontosabb fogalmakat ismertetjük

Ezek:

- a.) logikai szintek;
- b.) fan – out (F.O.);
- c.) fan – in (F.I.);
- d.) jelterjedési idő (t_{pd});
- e.) zavarvédetség;
- f.) disszipáció;
- g.) karakterisztikák.

a.) Az elektronikus digitális áramkörökben a 0 és 1 logikai értéket hordozó fizikai jellemző leggyakrabban a feszültség. A logikai szintekhez rendelt feszültség értékeket logikai szinteknek nevezzük. Az áramkörök kimenetén az alkatелеmek szórása és a változó környezeti feltételek (hőmérséklet, terhelés, tápfeszültség) miatt a logikai értékekhez tartozó feszültség kisebb-nagyobb mértékben különbözik. A logikai IGEN szint névleges értékét U_1 -gyel, a felső határát \overline{U}_1 - mal, alsó határát \underline{U}_1 -mal jelöljük. A NEM szint átlagos értékét U_0 -val, a felső határát \overline{U}_0 -mal, alsó határát pedig \underline{U}_0 -mal jelöljük. Megjegyezzük, hogy az aktív áramköröknél a bemeneten szélesebb tartományt engedünk meg, mint a kimeneten. Ez az áramkörök zavarvédetségét biztosítja. Az áramköri leírásokban a pozitívabb logikai feszültségszintet magas (high, H), a negatívabb feszültségszintet alacsony (low, L) szintnek is szokás nevezni. Pozitív logikáról beszélünk, ha az IGEN szintet a pozitívabb feszültséghez rendeljük, még negatív logika esetén az IGEN szintet negatívabb feszültség reprezentálja.

A gyakorlatban rendszerint a nagyobb abszolút értékű feszültségszinthez rendelik az 1 logikai értéket. A logikai 0-át többnyire a 0 V-hoz rendelik. A logikai értéket meghatározott értékű feszültségtartomány ill. meghatározott amplitúdójú impulzus jelenléte vagy hiánya reprezentálhatja. Előbbit sztatikus, utóbbit dinamikus megjelenítési formának nevezzük.

b.) Mivel az egyes logikai egységek bemeneti impedanciája véges értékű, a bemenetek a vezérlő fokozat számára terhelést jelentenek. Megállapodászerűen a meghajtó áramkör kimenetébe befolyó áramot pozitív irányú, míg a meghajtó áramkör kimenetéből a terhelésbe folyó áramot negatív irányú terhelő áramnak tekintjük. A digitális áramkörökben



az egyes egységek bemenetei hasonló felépítésűek, így azonos nagyságú terhelést jelentenek.

Ezt a leggyakrabban előforduló áramértéket egységterhelésnek nevezzük. Az egységek kimenetei az előírt specifikációt csak megengedett nagyságú terhelő áramok esetén teljesítik.

A megengedett terhelő áram és az egységterhelés áramának hányadosát dc. fan – out-nak (egyenáramú terhelhetőség) nevezzük. A dc. fan-out tehát azt adja meg, hogy egy kimenet hány bemenetet vezérelhet. A bemenetek terhelő kapacitása a meghajtó áramkörök jelterjedési idejét növeli. Ezért néhány áramköri rendszerben külön definiálják az un. ac. fan – out értékét, mely

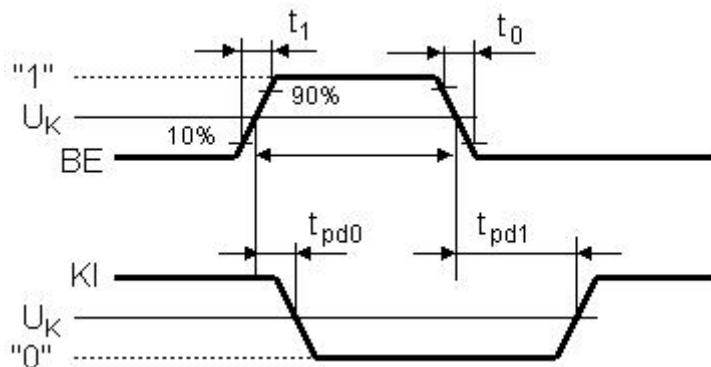
megadja azon vezérlő bemenetek számát, amelynél a specifikált kapcsolási idők még biztosan teljesülnek

c.) Fan-in alatt azt a számot értjük, amely megadja, hogy hány különböző információs bemenet áll rendelkezésre a kimenet befolyásolására.

d.) A digitális áramkörök a logikai funkciókat egy bizonyos időkéssel realizáljuk. Az áramkörök működési sebességéről a jelterjedési idő ad felvilágosítást. A terjedési idő a kimenet megváltozásának késése a változást előidéző bemeneti jelhez képest. A kimenet 1→0 változásának késési idejét lefutási késésnek mondjuk és t_{pd0} -val jelöljük. Hasonlóképpen a kimenet 0→1 változásának késését felfutási késésnek nevezzük és t_{pd1} -gyel jelöljük. Terjedési idő alatt a két késési idő átlagát értjük:

$$t_{pd} = \frac{t_{pd1} + t_{pd0}}{2}.$$

A logikai áramköröknél a terjedési időt a komparálási szint (U_K) elérésénél mérik. Komparálási szint alatt azt a feszültséget értik, amelynek elérésekor az áramkör átvált másik állapotba.



d.) A digitális berendezésekben alkalmazott áramkörök bemenetein nem kívánatos, un. zavaró jelek léphetnek fel. Ezek külső és belső eredetűek lehetnek. Ha a zavaró jelek amplitúdója olyan nagy, hogy az áramkör kimenetének állapota megváltozik anélkül, hogy a vezérlő áramkörök állapota megváltoznék, akkor hibás működés lép fel.

Az áramköröknek a zavaró jelekkel szembeni érzéketlenségét zavarvédettségnek nevezzük. A zavarvédettséget a hibás működést még éppen nem okozó zavaró jel amplitúdójával jellemezzük. Megkülönböztetünk sztatikus és dinamikus zavarvédettséget. Sztatikus zavaró jelek alatt az olyan zavaró jeleket értjük, melyek időtartama hosszabb mint a logikai áramkörökben a jelek átlagos terjedési ideje (t_{pd}). Tehát, ha a zavaró jel egyenáramú vagy igen lassan változik, akkor sztatikusnak tekintjük. A sztatikus zavarvédettség (zajimmunitás) tehát azt a járulékos feszültségszintet adja meg, amely a vizsgált áramkör bemenetét vezérlő jelszintre adható anélkül, hogy az áramkör kimenetén levő állapot emiatt megváltoznék.

A dinamikus zajvédettség esetén a t_{pd} időtartamnál rövidebb zavarjelek hatását vizsgáljuk. Jellemzésére a zavaró jel amplitúdójának és időtartamának szorzata használatos ($U_z \cdot t_z$).

f.) Disszipációnak azt a teljesítményt tekintjük, amely az áramkörben 50%-os kitöltési tényezőjű vezérlés mellett hővé alakul.

g.) A logikai áramkörök viselkedése jól követhető a különböző karakterisztikák alapján. Ilyenek: bemeneti, kimeneti és transzfer karakterisztika. A megadás legtöbbször grafikusán történik.

8. Kombinációs típusú funkcionális egységek

A félösszeadó kombinációs táblázata

mi	X _i	Y _i	S	C _y
0	0	0	0	0
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1

Bináris összeadás:

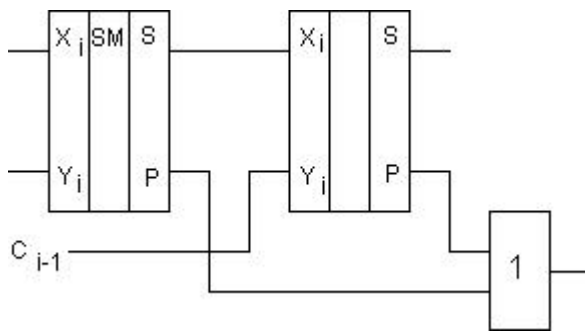
$$\sum_{1+1=0} C_y \quad 1$$

Bináris kivonás:

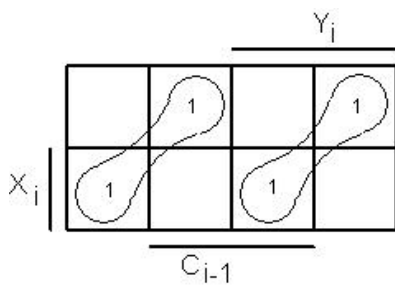
$$- C_y \quad 0-1=1 \quad 1$$

mi	X _i	Y _i	C _{i-1}	Σ	Carry	-	Carry
mi	X _i	Y _i	C _{i-1}	S	C _y	K	C _y
0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	1
2	0	1	0	1	0	1	1
3	0	1	1	0	1	0	1
4	1	0	0	1	0	1	0
5	1	0	1	0	1	0	0
6	1	1	0	0	1	0	0
7	1	1	1	1	1	1	1

A teljes összeadó/kivonó kombinációs táblázata

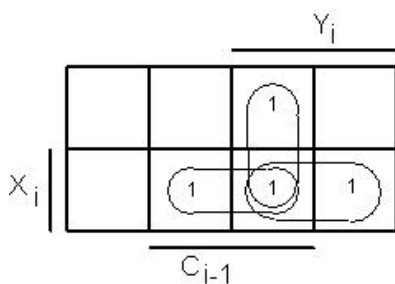


Összegfüggvény (teljes összeadó/kivonó):



$$S = X_i \oplus Y_i \oplus C_{i-1}$$

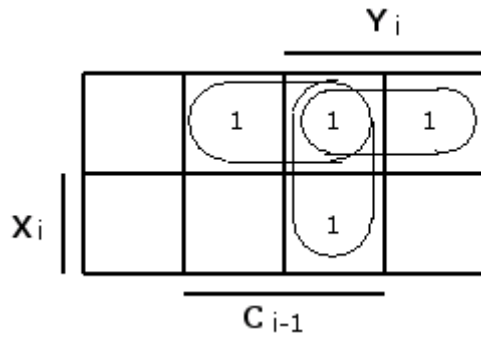
A teljes összeadó átvitelfüggvénye:



$$C_y = Y_i C_{i-1} + X_i C_{i-1} + X_i Y_i$$

A teljes kivonó átviteli függvénye:

$$C_Y = Y_i C_{i-1} + \bar{X}_i C_{i-1} + \bar{X}_i Y_i$$



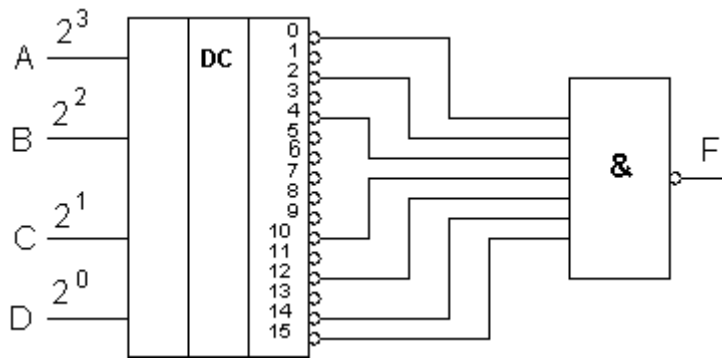
A félkivonó kombinációs táblázata

mi	X_i	Y_i	K	C_Y
0	0	0	0	0
1	0	1	1	1
2	1	0	1	0
3	1	1	0	0

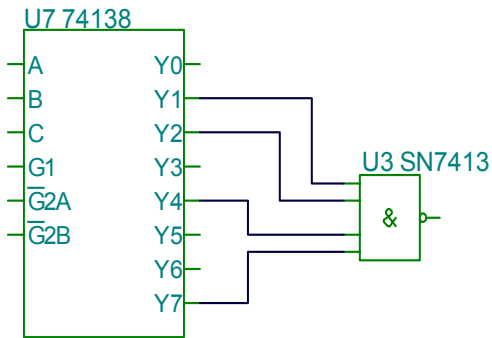
Dekódoló áramkör (DC) (minterm generátor, demultiplexer)

Példa: $F(A,B,C,D) = \Sigma(0,2,4,10,12,14,15)$

A fenti függvény realizálása DC áramkörrel:



Példa: a teljes összeadó S (Σ) összegfüggvényének megvalósítására DC áramkörrel.

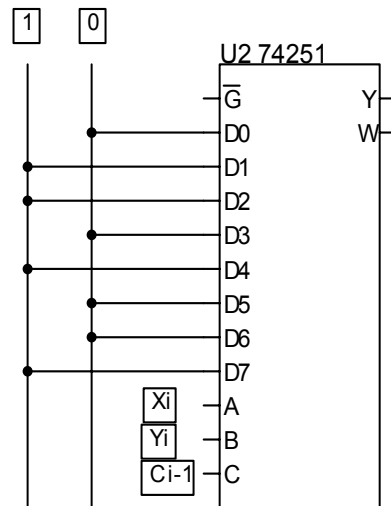


Adatszelektorok (vonalkiválasztók, multiplexerek **MUX**-ek)

Más elnevezéssel ULM- univerzális logikai modul

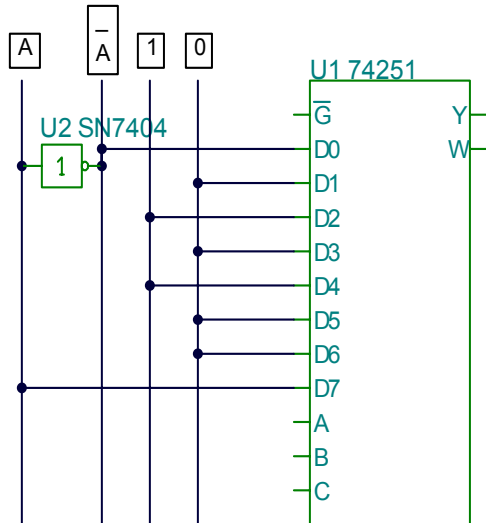
MUX: több bemenetű és 1 kimenetű kombinációs áramkör

Példa: a teljes összeadó S (Σ) összegfüggvényének megvalósítására MUX áramkörrel

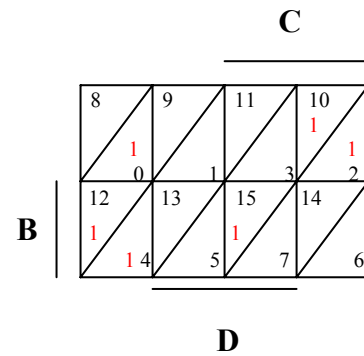


„n” bemenetű logikai függvény realizálása „n-1” bemenetű MUX-rel

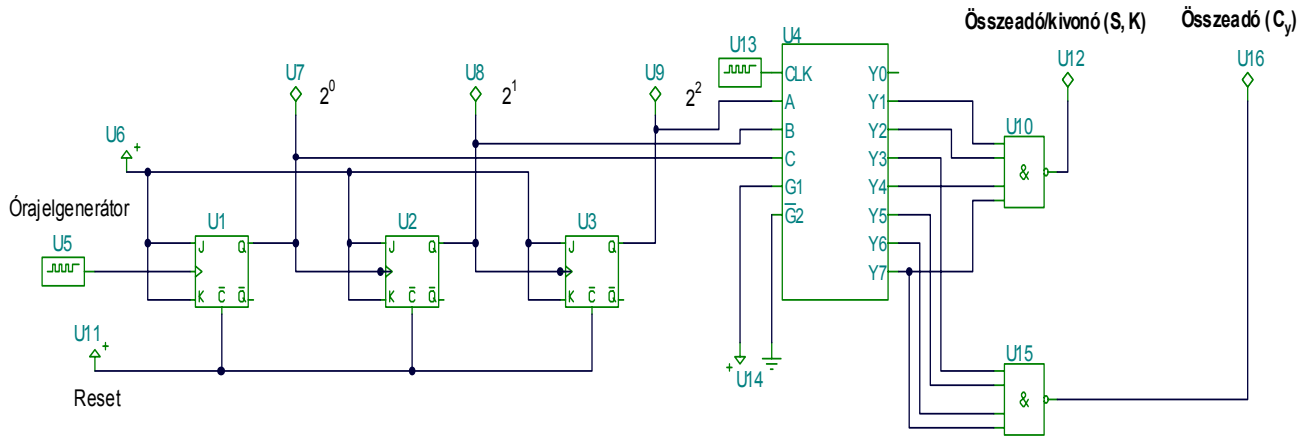
Példa: realizáljuk három bemenetű multiplexerrel (MUX) az $F(A,B,C,D)=\Sigma(0,2,4,10,12,15)$ négyváltozós logikai függvényt!



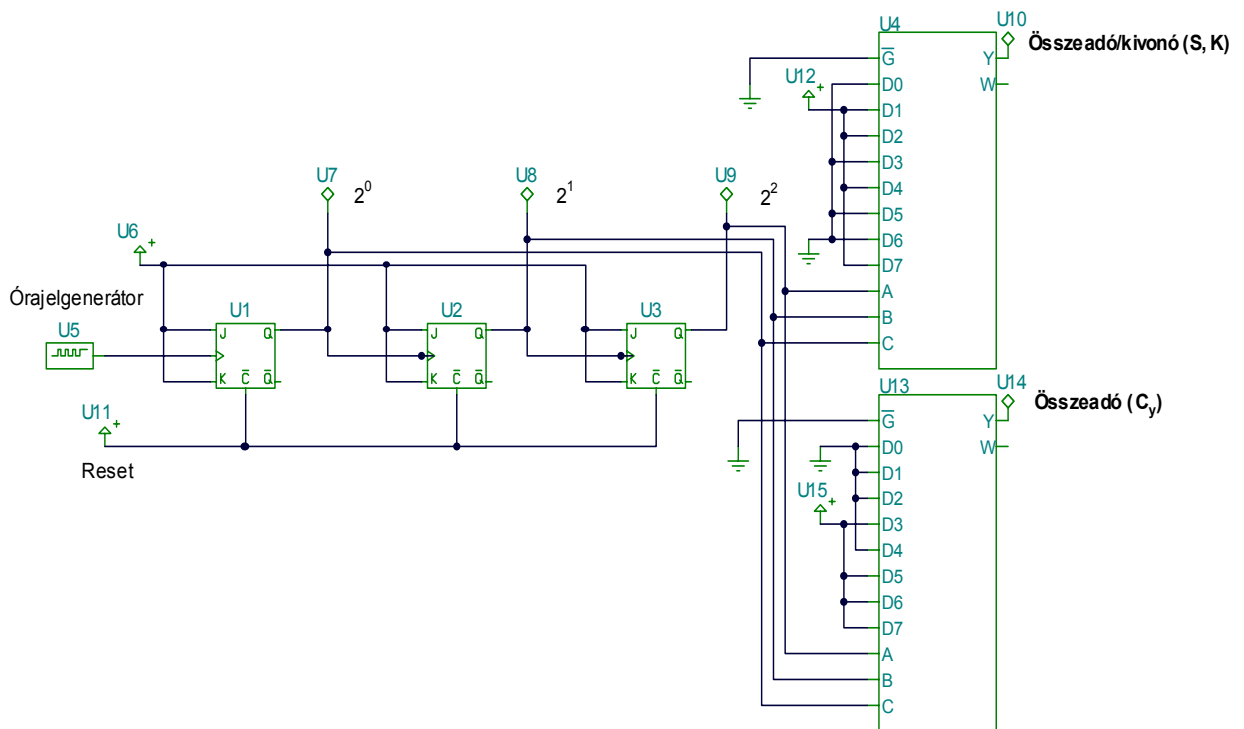
$2^1 m_i$	$2^0 m_i$	
0	0	„0”
0	1	\overline{A}
1	0	A
1	1	1



A teljes összeadó/kivonó (S, K) függvényeinek megvalósítása DC-re



A teljes összeadó/kivonó (S, K) függvényeinek megvalósítása MUX-re



A fenti ábrák lehetőséget biztosítanak, a teljes összeadó/kivonó (S, K) és a teljes összeadó C_y függvényeinek működés közben, szimulációs úton való nyomon követésére. A fenti hálózat DC, a lenti pedig MUX áramkörökkel került kivitelezésre.

9. Tárolóelemek, F.F.-ok

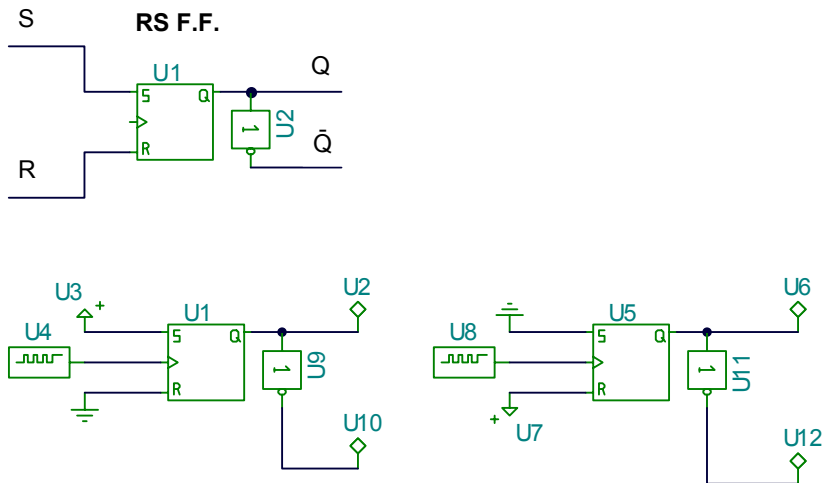
F.F.: 1 bit információ tárolására alkalmas elemi memória cella.

RS F.F.

Más elnevezése bistabil multivibrátor

S: beírás

R: törlés

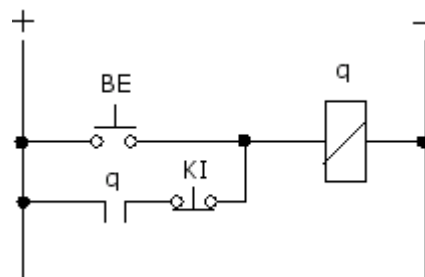
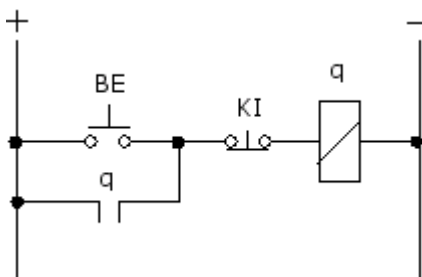


A fenti ábrán lévő RS1 F.F., működése közben is követhető.

Törlésre elsőbbséget biztosító kapcsolás RS0, beírásra elsőbbséget biztosító kapcsolás RS1 F.F.

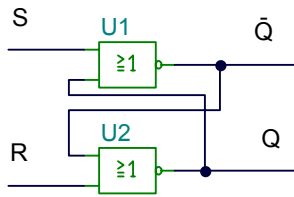
Elengedésre elsőbbséget biztosító relés
alapkapsolás (RS0 F.F.)

Meghúzásra elsőbbséget biztosító relés
alapkapsolás (RS1 F.F.)

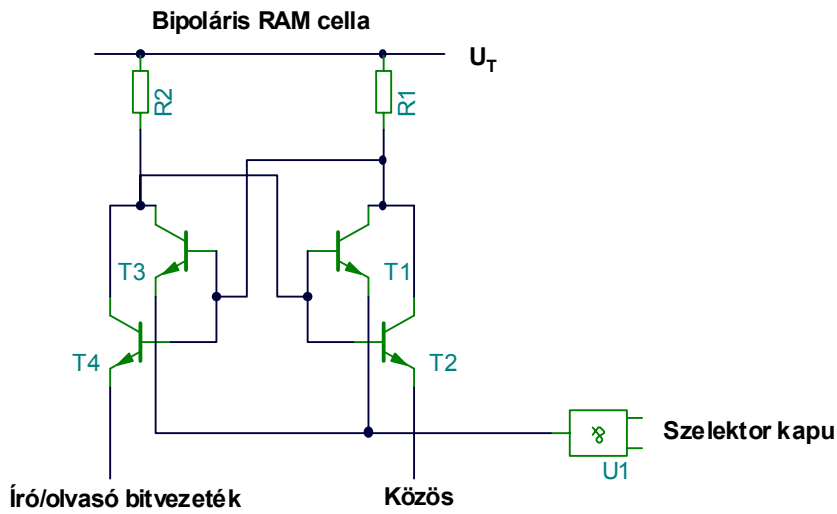
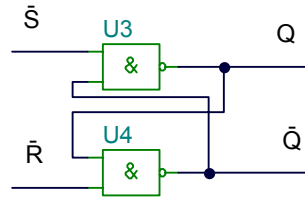


TTL elemekkel realizált RS F.F.-ok

TTL NOR elemekkel realizált RS1 F.F.

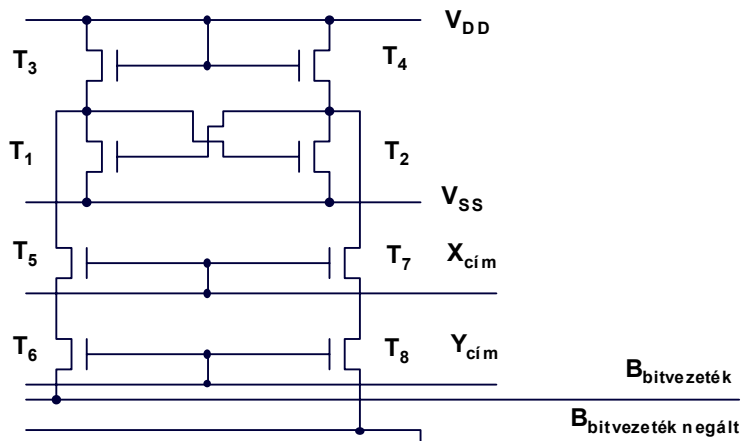


TTL NAND elemekkel realizált RS0 F.F.



A fenti ábra bipoláris tranzisztorokkal felépített bistabil (RS F.F.) multivibrátort szemléltet.

Nyolc tranzisztorral felépített sztatikus MOS RAM cella



A fenti ábra MOS tranzisztorokkal felépített RAM cella.

JK F.F.

Példaként számlálók (aszinkron, szinkron), SHIFT regiszterek készítésére alkalmazható JK F.F. 1- 1 vezérlés esetén komplementál (állapotot vált)

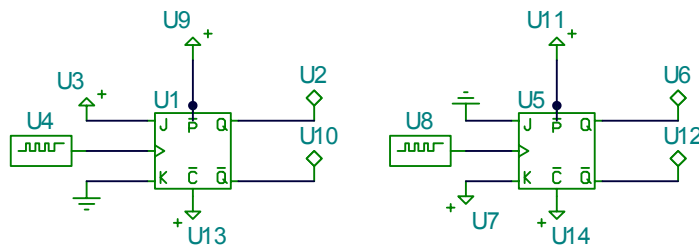
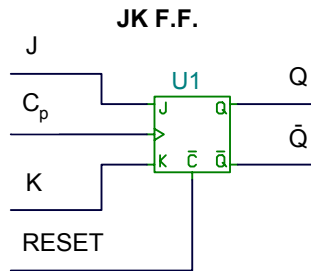
J: beírás

K: törlés

C_p : órajel

R, C: Reset

P: Preset



A fenti ábra a JK F.F.-ot szemlélteti működés közben.

D F.F.

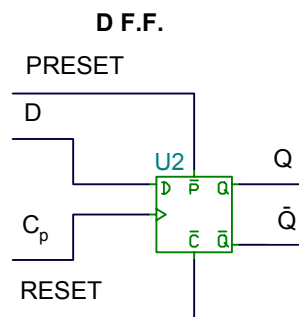
Példaként SHIFT regiszterek készítésére alkalmazzák a D F.F.-okat

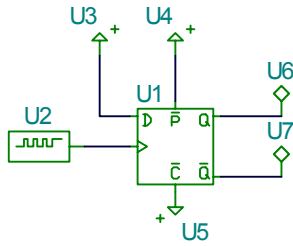
D: beírás

C_p : órajel

R, C: Reset

P: Preset

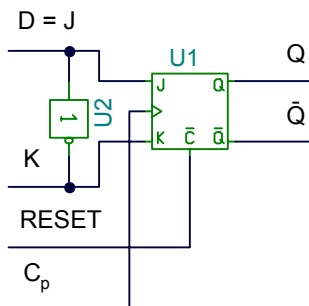




A fenti ábra a D F.F.-ot ábrázolja működés közben.

Példaként léptető (SHIFT) regiszterek készítéséhez is alkalmazható JK F.F.-ból D F.F.

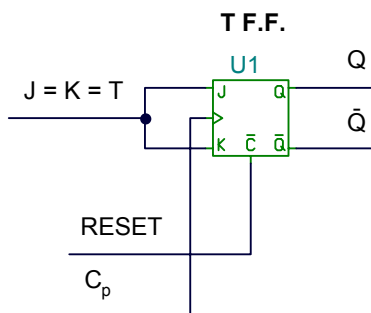
JK F.F.-ból készített D F.F.



T F.F.

T: komplementáló F.F.

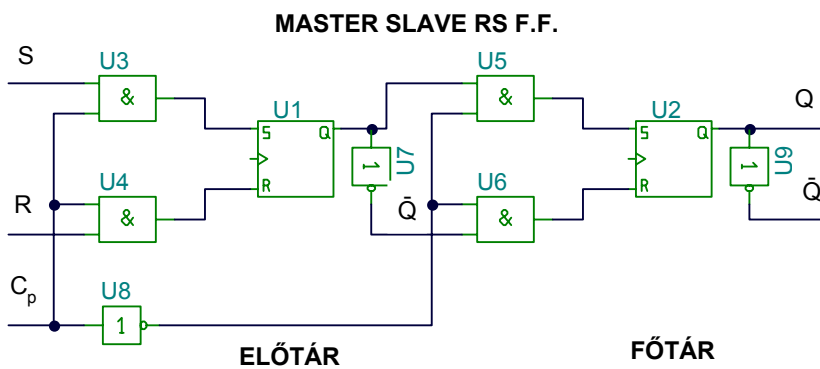
$J=K=T$ helyettesítéssel készíthető JK F.F.-ból



Q_n	Q_{n+1}	R	S	J	K	D	T
0	0	x	0	0	x	0	0
0	1	0	1	1	x	1	1
1	0	1	0	x	1	0	1
1	1	0	x	x	0	1	0

A fenti táblázat, az RS; JK; D és T F.F.-ok vezérlési tábláit tartalmazza, a beírt „0”, „1” és „x” szimbólumok alapján követhetők a F.F.-ok működése.

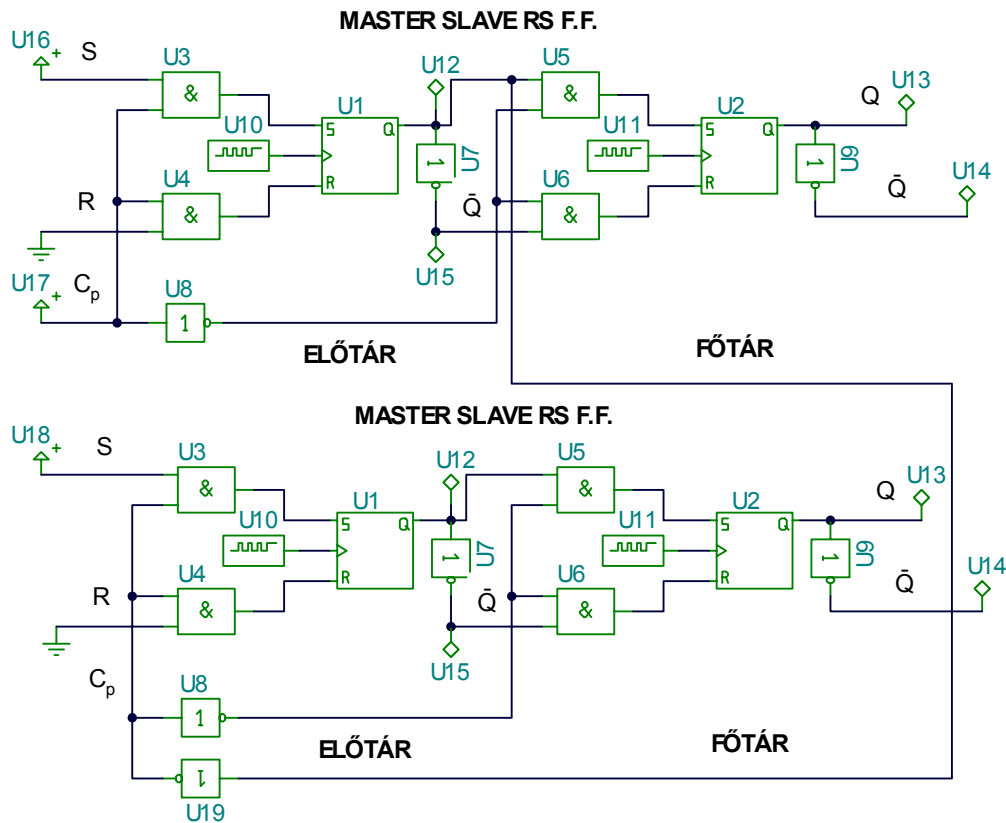
Master - slave RS F.F.



A Master-Slave RS F.F.-ok, közbenső tárolás szinkron működésű tároló elemek.
A fenti ábrából kitűnik, hogy 2 db szinkron RS tárolóból és egy inverterből állnak.

Működése:

- $C_p = 0$ esetében az S és R bemenetek le vannak választva a Master tárolóról;
- $C_p = 1$ esetében az S és R bemenetekre adott sztatikus jel beíródik a Master fokozatba;
- ezt követően a Slave F.F. bemenete az inverteren keresztül tiltott.



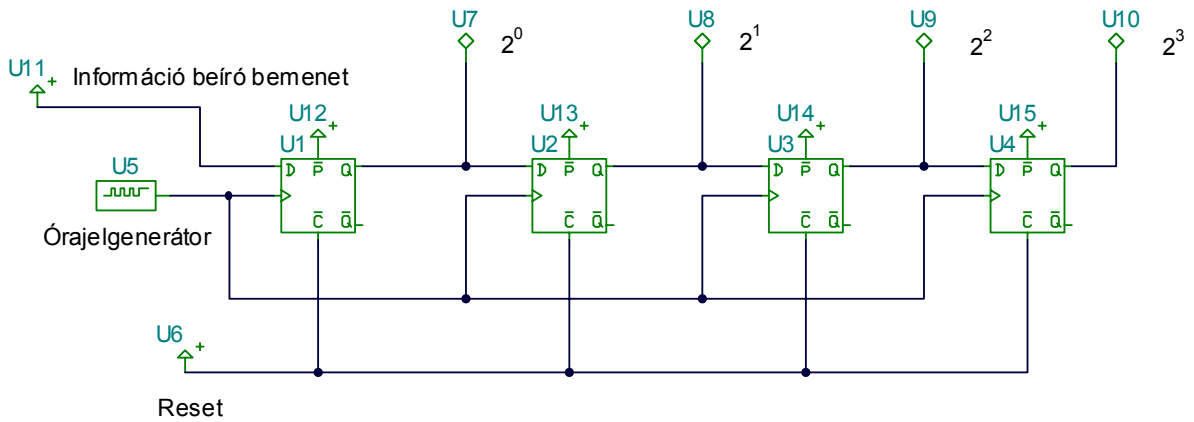
A fenti ábrán lévő Master-Slave RS F.F. működés közben is nyomon követhető.

Megjegyzés:

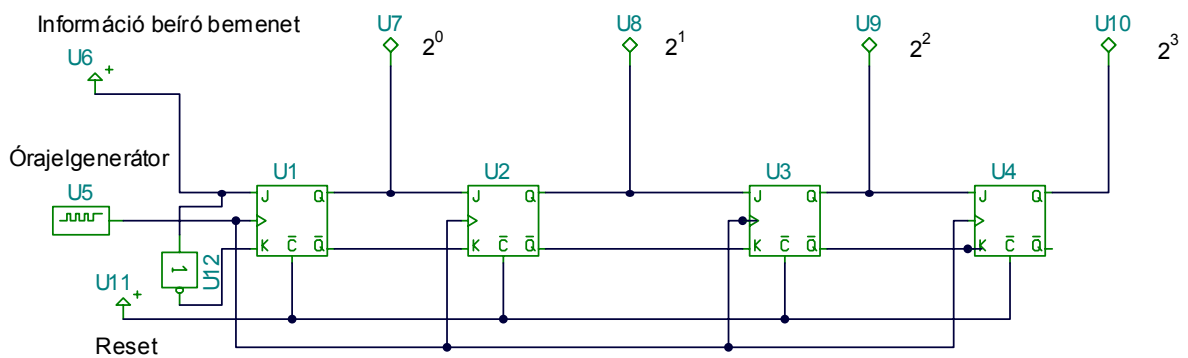
Az ábrán lévő U19 jelű inverter természetesen nem a Master-Slave RS F.F. áramköri tartozéka, hanem a C_p -vel („órjel”) jelölt inverter bemenetén, a 0; 1 logikai jeleket automatizálja, a jobb megértés érdekében.

10. Léptetőregiszterek

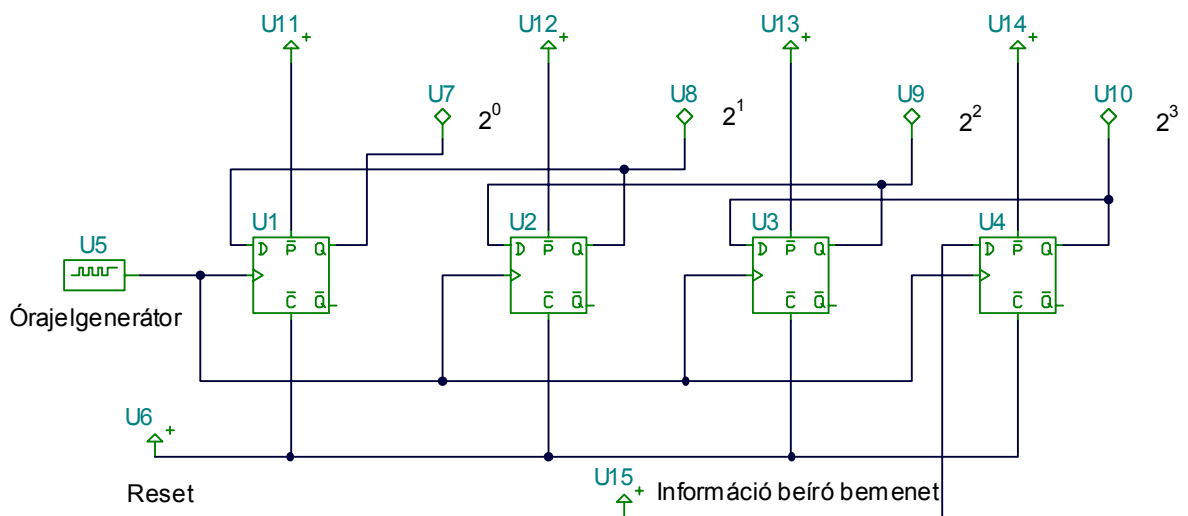
4 bites jobbra vagy előreléptető SHIFT regiszter felépítése D F.F.-okkal.
Szinkron üzemmód, mert az órajel közös, a F.F.-ok egyszerre billennek.



4 bites jobbra léptető regiszter JK F.F.-ból kialakított D F.F.-okkal felépítve.



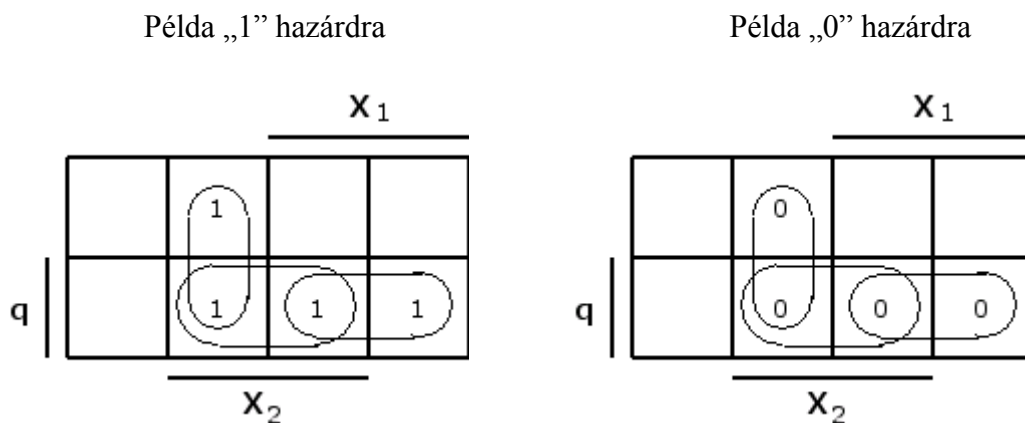
4 bites balra vagy visszaléptető SHIFT regiszter kialakítása D F.F.-okkal.



11. Hazárd, versenyhelyzet

Valahányszor jelváltozás lép fel a bemeneteken, fennáll annak a veszélye, hogy a kimeneteken átmenetileg tranziens jelenség, azaz hamis működésű jel lép fel. Ebben az esetben a kimenő jelek jelkombinációi hamis állapotokat mutatnak. Aszinkron szekvenciális hálózatok gerjesztési függvényeiben fellépő olyan jelenség, ami hamis működést eredményez, azt hazárdnak nevezzük. A gerjesztési függvényekben előforduló statikus hazárdokat lényeges hazárdnak hívjuk. Előfordulhatnak statikus „1”, ill. statikus „0” hazárdok, amelyek „1”-el jelölt celláról szomszédos „1”-el jelölt, vagy „0”-val jelölt celláról szomszédos „0”-val jelölt cellára történő átmenetek esetében fordulnak elő. A dinamikus hazárdok „0”-val jelölt celláról „1”-el jelölt cellára és fordítva történő átmenetek eseteiben jelennek meg. A dinamikus hazárdok többfokozatú hálózatokban fordulnak elő. Kétszintű ÉS/VAGY hálózatok dinamikus hazárdot nem tartalmaznak. Versenyhelyzetek alakulnak ki aszinkron sorrendi hálózatok gerjesztési függvényeiben, a belső állapotok megváltozásakor abban az esetben, ha egyidejűleg egynél több szekunder elem (F.F.) szeretné az állapotát megváltoztatni. Más szavakkal megfogalmazva, ha az egymás után következő állapot kódja közötti Hamming távolság $D_{AB} \geq 2$. Kritikus a versenyhelyzet, ha a végső stabil állapot függ az állapotváltozások sorrendjétől.

Statikus „1”, ill. statikus „0” hazárd észrevétele a gerjesztési táblákban, a F.F.-ok vezérlési függvényeiben úgy történik, hogy két szomszédos „1”-el vagy „0”-val jelölt mintermek vagy maxtermek nincsenek közös tömbbel lefedve. Az „1” hazárd újabb ÉS elem alkalmazásával, a „0” hazárd pedig újabb VAGY elem alkalmazásával küszöbölhető ki.



Hazárdmentesített függvények:

$$Q = q x_1 + \bar{x}_1 x_2 + q x_2$$

$$Q = (\bar{q} + \bar{x}_1) \& (x_1 + \bar{x}_2) \& (\bar{q} + \bar{x}_2)$$

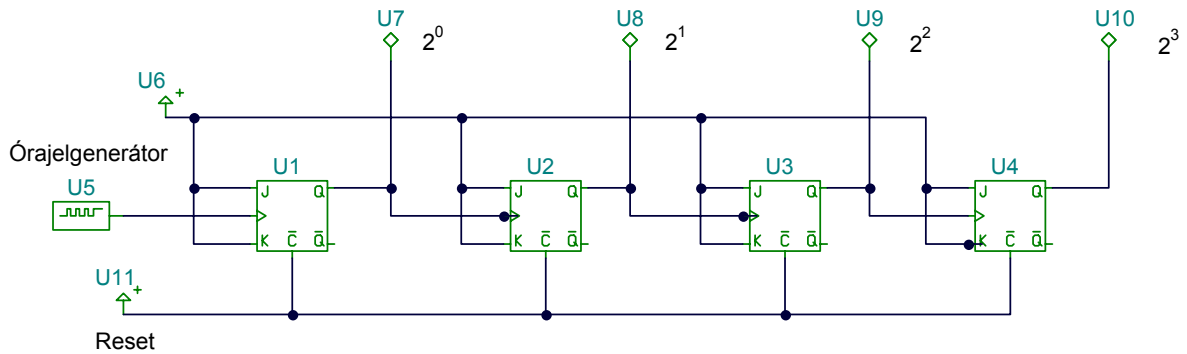
Az „1” hazárd újabb ÉS, a „0” hazárd pedig újabb VAGY elem alkalmazásával küszöbölhető ki.

Alapállapotban „1” hazárd található volt az 5, 7 mintermekben, valamint „0” hazárd a 0, 2 maxtermekben.

12. Aszinkron számlálók felépítése, készítésük és kialakításuk

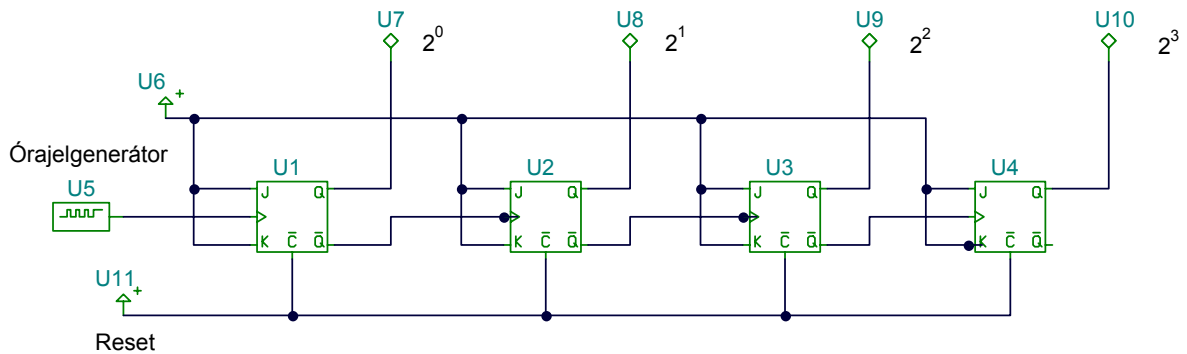
- a F.F.-ok egymást billentik, nincs közösített órajelük
- a 2^0 helyiértékű F.F. Q kimenete előszámlálók esetében, a 2^1 helyiértékű F.F. órajel bemenetére csatlakozik
- visszaszámlálónál a 2^0 helyiértékű F.F. \bar{Q} kimenete csatlakozik, a 2^1 helyiértékű F.F. órajel bemenetére

4 bites aszinkron előszámláló felépítése JK F.F.-okkal.



A közösített JK bemenetet tápfeszültségre kell kötni, így a JK F.F.-ból T F.F. készül, $J=K=T$ helyettesítéssel.

4 bites aszinkron visszaszámláló kialakítása JK F.F.-okkal.



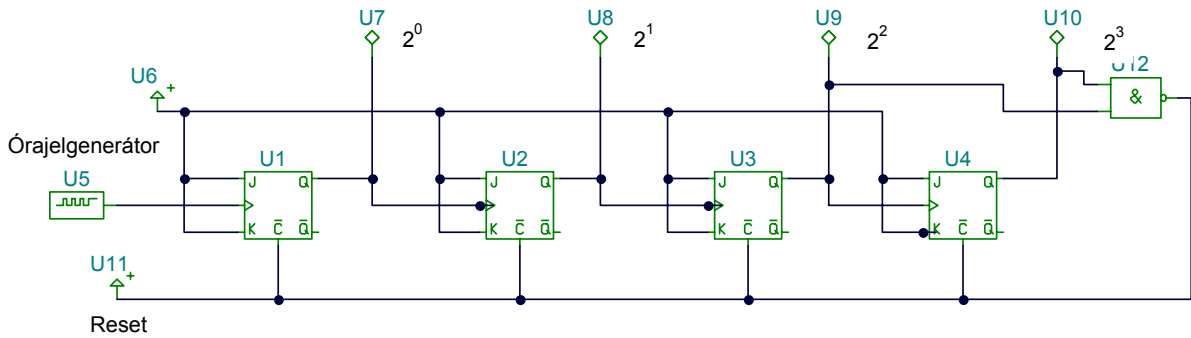
Különleges aszinkron számlálók készítése

Fix hosszúságú ismétlődő ciklusú aszinkron bináris előre ill. visszaszámláló készítése JK F.F.-okkal

Készítés menete:

$N=12$ esetére

- a F.F.-ok száma $\geq \log_2 N$
- elkészítjük a kívánt irányú aszinkron számlálót
- N bináris 1-eseit dekódoljuk egy NAND elemmel, majd a NAND kimenetét, a közösített Resetre kötjük

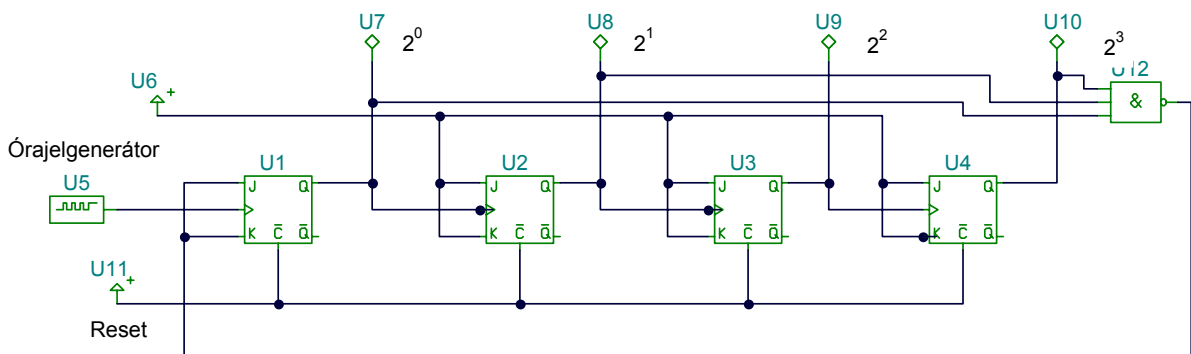


A fenti ábra a számláló felépítését mutatja.

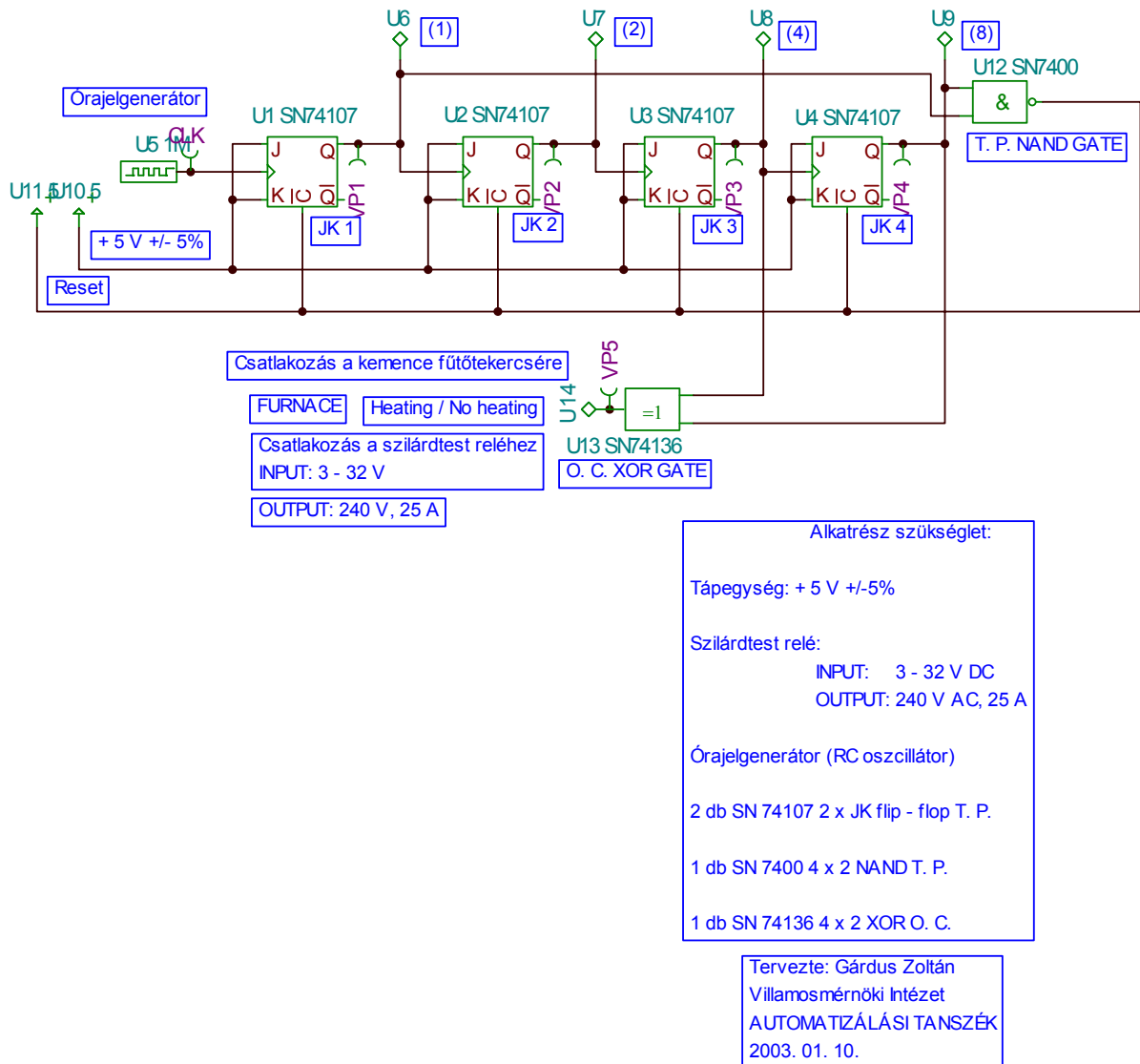
N=12 esetére fix hosszúságú, leálló aszinkron számláló készítése JK F.F.-okkal

Készítés menete:

- a F.F. száma $\geq \log_2 N$
- N-1 állapot dekódolása egy NAND elemmel, majd kimenetét, a 2^0 helyiértékű F.F. közösített JK F.F. bemenetére kötjük.



JK F.F.-okból kialakított, aszinkron számlálóval megvalósított fűtőelem ki/bekapcsolása.

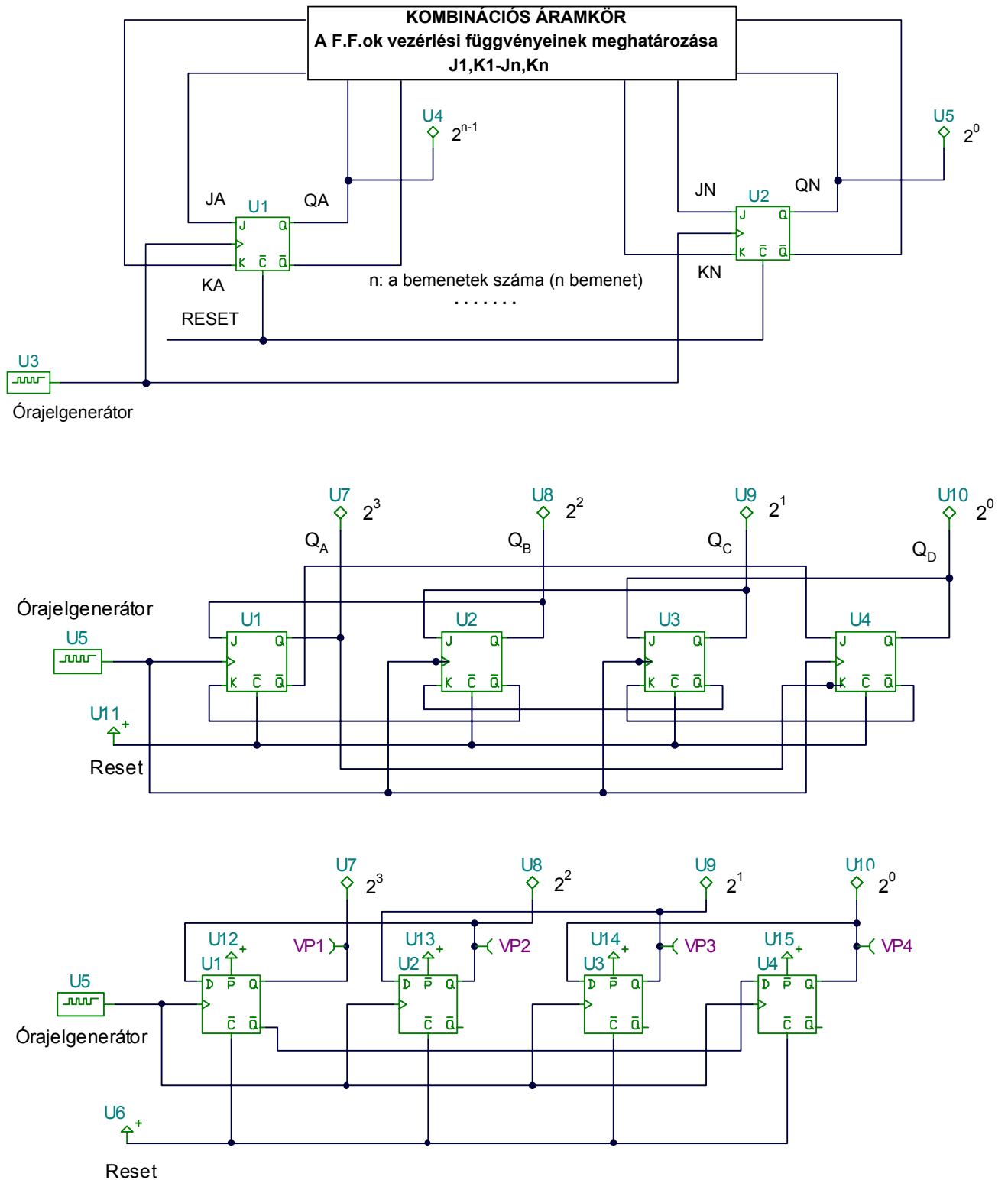


A fenti ábra 4 órajel impulzusig „0”, a következő 4 órajel impulzusig pedig „1” jelet szolgáltat. A számláló alkalmas a későbbiekben leírtak szerint, az **Analízis/Digitális nyomkövetés** menüpontban az idődiagram megjelenítésére.

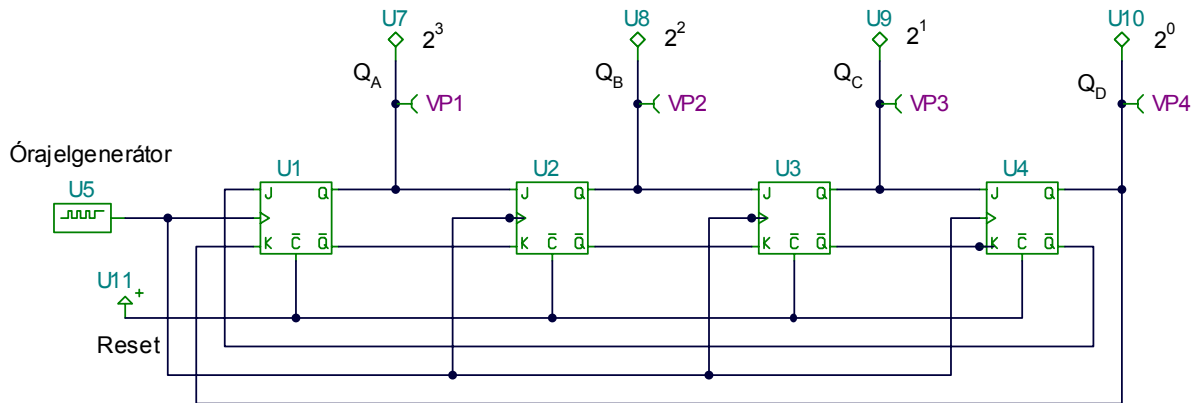
A fenti hálózat komplett alkatrészszükségletet is tartalmaz a megvalósításhoz.

13. Szinkron számlálók tervezése, felépítése

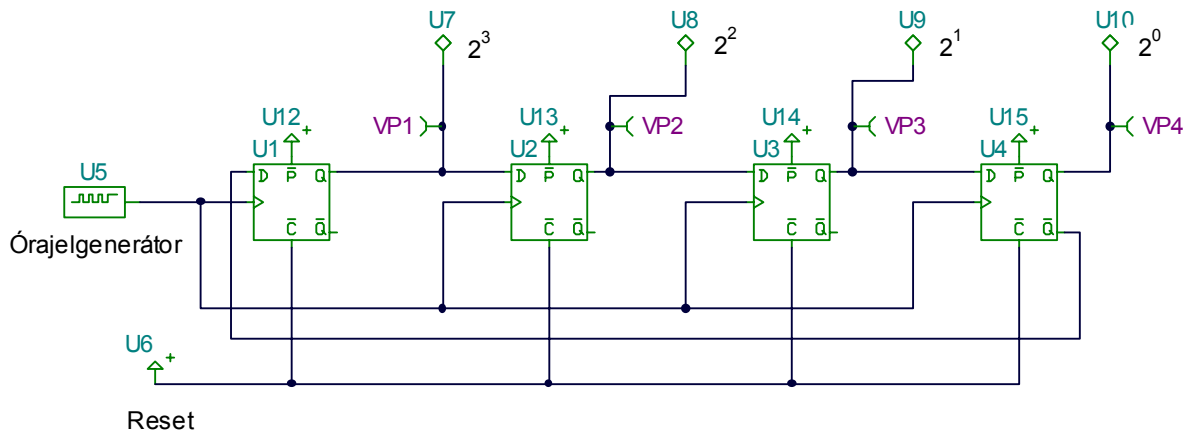
A szinkron számlálók felépítésének sematikus modellje



A fenti ábrák 4 bites JOHNSON szinkron előreszámlálók kialakítását szemléltetik, JK ill. D F.F.-okból felépítve.



Az ábra 4 bites JOHNSON szinkron visszaszámlálót szemléltet, ami JK F.F.-okból lett kialakítva, valamint alkalmas az idődiagram megjelenítésére.



A fenti ábra szintén 4 bites JOHNSON szinkron visszaszámlálót ábrázol, de D F.F.-okból felépítve, ami szintén alkalmas az idődiagram megjelenítésére.

Az alábbi feladat megoldása segítséget nyújt, a szinkron számlálók tervezéséhez.

Ezt követően mintaprogramot láthatunk JOHNSON számláló programozásához 8085 μ p-ral, ahol a számlálандó impulzusok időzítése (órajelgenerátor) négy alternatíva szerint került kidolgozásra.

Feladat: tervezzon 4 bites JOHNSON szinkron előreszámláló JK F.F.-okkal!

A tervezés menete:

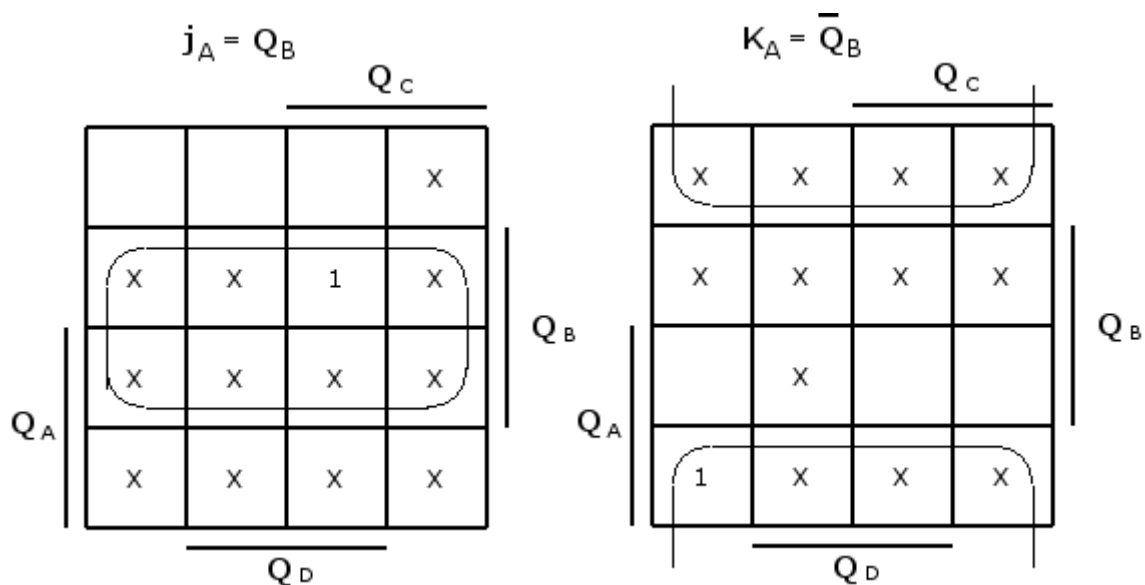
1. kombinációs táblázat felvétele;
2. a F.F.-ok vezérlési tábláinak a felvétele;
3. a F.F.-ok vezérlési függvényeinek meghatározása a vezérlési táblákból;
4. realizálás.

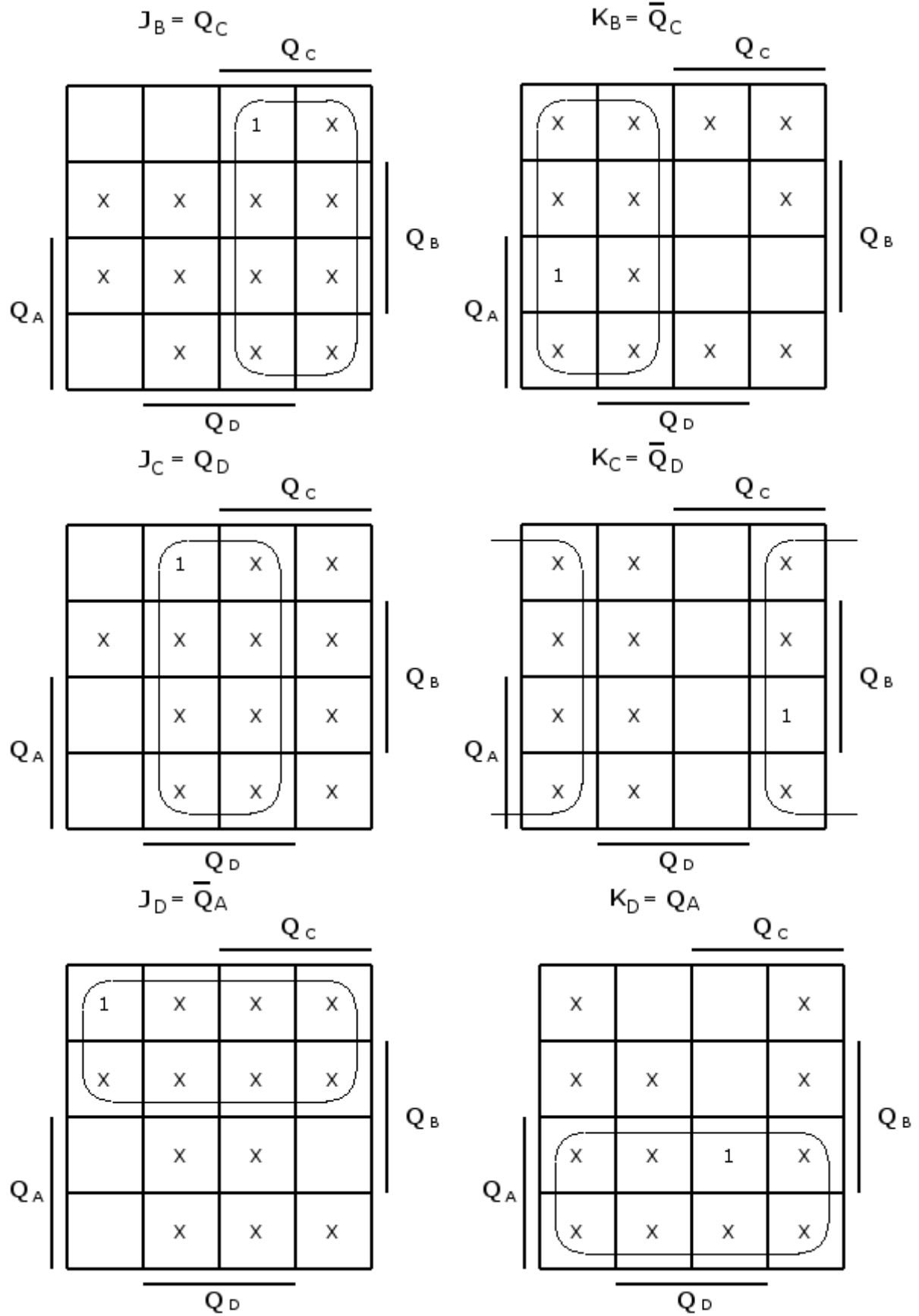
Kombinációs táblázat

m_i	inf	Q_{An}	Q_{Bn}	Q_{Cn}	Q_{Dn}	Q_{An+1}	Q_{Bn+1}	Q_{Cn+1}	Q_{Dn+1}
0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	1	0	0	1	1
3	2	0	0	1	1	0	1	1	1
7	3	0	1	1	1	1	1	1	1
15	4	1	1	1	1	1	1	1	0
14	5	1	1	1	0	1	1	0	0
12	6	1	1	0	0	1	0	0	0
8	7	1	0	0	0	0	0	0	0

J_A	K_A	J_B	K_B	J_C	K_C	J_D	K_D
0	X	0	X	0	X	1	X
0	X	0	X	1	X	X	0
0	X	1	X	X	0	X	0
1	X	X	0	X	0	X	0
X	0	X	0	X	0	X	1
X	0	X	0	X	1	0	X
X	0	X	1	0	X	0	X
X	1	0	X	0	X	0	X

A F.F.-ok vezérlési függvényeinek a meghatározása a KV táblákból történik, az alábbiak szerint:





Példa: 4 bites JOHNSON kódú számláló készítésére 8085-ös mikroprocesszorral

1. fix késleltetéssel
2. változtatható késleltetési idővel
3. a számlálás nyomógomb megnyomására történik
4. órajel generátorként egy INPUT porton elhelyezkedő RC oszcillátort használunk fel

A késleltetések szubrutinban helyezkedjenek el, mely a főprogram után áll, a szubrutint a JOHNSON kódnak megfelelő memória táblázat követi:

00h
01h
03h
07h
0Fh
0Eh
0Ch
08h

A főprogram a 3000h címtől kezdődődjön

```

                LXI SP,20C2h
                MVI A,80h
                OUT C3h
hurok1:        LXI H, memória táblázat kezdő címe
                MOV A,M
                OUT C2h
                CALL rutin
hurok:         INX H
                MOV A,M
                OUT C2h
                CALL rutin
                CPI 08h
                JZ hurok1
rutin:         JMP hurok

```

1. fix késleltetési idő szubrutin

```

rutin:         MOV B,A
                LXI D, FFAAh
hurok2:        DCX D
                JNX5 hurok2
                MOV A,B
                RET

```

2. az E1h porton lévő peremkerekessel kapcsoló szolgáltatása a késleltetési időt, a szubrutinban

```

rutin:         MOV B,A
                IN E1h

```



```

MOV D,A
MVI E,00h
hurok2: DCX D
        JNX5 hurok2
        MOV A,B
        RET

```

3. az E2h port 0-adik bitjén lévő nyomógomb megnyomása aktivizálja a számláló léptetését, amit szubrutinban helyezünk el

```

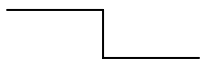
rutin:   MOV B,A
hurok2:  IN E2h
        ANI 01h
        CPI 01h
        JZ hurok3
        JMP hurok2
hurok3:  LXI D,FFAAh
hurok4:  DCX D
        JNX5 hurok4
        MOV A,B
        RET

```

4. Órajel generátorként az E2h porton lévő RC oszcillátort használjuk szubrutinban

Az E2h port 4.,5. bitje tartalmazza az RC oszcillátort. Abban az esetben, ha az 5. bit értéke logikai „1”-es és a 4. bit értéke logikai „0” működjön a számláló!

7 6 5 4 3 2 1 0



```

rutin:   MOV B,A
hurok2:  IN E2h
        ANI 30h
        CPI 20h
        JZ hurok3
        JMP hurok2
hurok3:  IN E2h
        ANI 30h
        CPI 00h
        JZ hurok4
        JMP hurok3
hurok4:  MOV A,B
        RET

```

A főprogram lefordítása gépi kódra

3000: 31
C2
20
3E
80
D3
C3
3007: 21
29
30
7E
D3
C2
CD
1F
30
3010: 23
7E
D3
C2
CD
1F
30
FE
08
CA
07
30
C3
10
30 a főprogram vége

1. számú szubrutin lefordítása gépi kódra

301F: 47
11
AA
FF
3123: 1B
DD
23
30
78
C9 visszatérés a főprogramba a szubrutinból

2. számú szubrutin lefordítása gépi kódra

301F: 47
DB
E1
57
1E
00
3025: 1B
DD
25
30
78
C9 visszatérés a főprogramba a szubrutinból

3. számú szubrutin lefordítása gépi kódra

301F: 47
3020: DB
E2
E6
01
FE
01
CA
2C
30
C3
20
30
302C: 11
AA
FF
302F: 1B
DD
2F
30
78
C9 visszatérés a főprogramba a szubrutinból

4. számú szubrutin lefordítása gépi kódra

301F: 47
3020: D3
E2
E6
30
FE
20
CA

2C
30
C3
20
30
302C: DB
E2
E6
30
FE
00
CA
38
30
C3
2C
30
3038: 78
C9 visszatérés a főprogramba a szubrutinból

14. Aszinkron sorrendi (szekvenciális) hálózatok

Aszinkron sorrendi hálózatok tervezési filozófiája

Aszinkron sorrendi hálózat: ugyanazon bemeneti kombinációkhoz más-más kimeneti események tartoznak, de tartozhatnak ugyanazok is, a bemeneti változók kombinációinak bekövetkezési sorrendjétől szigorúan függően. A F.F.-ok egymást billentik a bemeneti feltétel teljesülését követően.

Tervezés menete:

- elsődleges állapotábra felvétele;
- ekvivalens állapotok keresése;
- redukált tábla elkészítése;
- gerjesztési táblák felvétele;
- gerjesztési függvények meghatározása;
- vezérlési táblák felvétele;
- vezérlési függvények meghatározása;
- kimeneti táblák felvétele;
- kimeneti függvények meghatározása;
- realizálás (relé, IC, PLC, mikroprocesszor, FPGA, stb).

Kimeneti függvények meghatározása

A kimeneti függvényeket a kimeneti táblák felvétele után határozzuk meg.

A kimeneti táblák száma megegyezik a kimeneti függvények számával.

A kimeneti táblák mérete, peremezése, elrendezése megegyezik a redukált tábla méretével, elrendezésével és peremezésével.

A kimeneti táblák kitöltése az alábbi módon történik:

- egyszerre nézzük az elsődleges és a redukált táblát;
- a stabil állapothoz tartozó kimeneti függvények értékét a stabil állapot cellájába írjuk (lehet 0 vagy 1);
- az instabil állapothoz tartozó kimeneti függvények értékét az instabil állapot cellájába írjuk és redundánsként kezeljük.

1. Feladat:

Z1 szelep 3x feltölt, majd Z2 1x és minden feltöltés után Z3 szelepen keresztül leürítés (Z1/Z2 =3/1). A rendszer kiinduláskor legyen alapállapotban!.

Az fenti feladat tervezésének lépései:

1. elsődleges állapotábra felvétele;
2. ekvivalens állapotok keresése;
3. redukált tábla felvétele;
4. állapot gráf felvétele;
5. F.F. -ok számának meghatározása;
6. állapotkódolás;
7. F.F. -ok vezérlési és gerjesztési függvényeinek meghatározása;
8. kimeneti táblák felvétele;

9. kimeneti függvények meghatározása;
10. realizálás.

1. Elsődleges tábla felvétele:

Az un. egy átmenetű GRAY-kódot fogjuk használni, a versenyhelyzet mentes hálózat kialakítása érdekében az állapotkódolásra, a gerjesztési táblákban.

Xa, Xf : bemenetek, Z1, Z2, Z3:kimenetek

00	01	11	10	Z1	Z2	Z3
a	b		-	1	0	0
	b	c	-	1	0	0
	d	c	-	0	0	1
e	d		-	0	0	1
e	f		-	1	0	0
	f	g	-	1	0	0
	h	g	-	0	0	1
i	h		-	0	0	1
i	j		-	1	0	0
	j	k	-	1	0	0
	l	k	-	0	0	1
m	l		-	0	0	1
m	n		-	0	1	0
	n	o	-	0	1	0
	p	o	-	0	0	1
a	p		-	0	0	1

2. Redukált tábla és az állapot gráf felvétele:

- a redukált tábla kialakítása az ekvivalens állapotok keresésével történik.

A redukált táblával meg fog egyezni a gerjesztési tábla (k), a vezérlési tábla (k) ill. a kimeneti tábla (k) mérete, peremezése, elrendezése.

a	b	c	-
e	d	c	-
e	f	g	-
i	h	g	-
i	j	k	-
m	l	k	-
n	n	o	-
a	p	o	-

0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

A $\log_2 s$ összefüggéssel meghatározható a F.F.-ok száma, ahol s a sorok számát jelöli. Mivel 8 sor található a redukált táblában, így a F.F.-ok száma 3 db.

2^4	2^3	2^2	2^1	2^0
q_1	q_2	q_3	Xa	Xf

A fenti táblázat a redukált tábla, gerjesztési táblák, vezérlési táblák és a kimeneti táblák (5 bemeneti változós) peremezését helyiérték szerint mutatja.

3. Gerjesztési tábla felvétele, amely egyidejűleg tartalmazza a 3 F.F. gerjesztését (Q_1, Q_2, Q_3):

000	000	001	-
011	001	001	-
011	011	010	-
110	010	010	-
110	110	111	-
101	111	111	-
101	101	100	-
000	100	100	-

Az RS F.F. vezérlési táblája:

Q_n	Q_{n+1}	R	S	Funkció
0	0	x	0	Tárol
0	1	0	1	Beír
1	0	1	0	Töröl
1	1	0	x	Tiltott

Q1 tábla felvétele:

0	0	0	-
0	0	0	-
0	0	0	-
1	0	0	-
1	1	1	-
1	1	1	-
1	1	1	-
0	1	1	-

$$Q_1 = q_1 q_2 + q_1 q_3 + q_1 X a + q_2 q_3 \bar{X} a$$

A gerjesztési függvényből (Q1) a meghúzási kombináció (S1) kiolvasható, mégpedig az a kombináció, amely nem tartalmazza a q1-et.

$$S1 = q_2 q_3 \bar{X}_a$$

R1 tábla felvétele az RS F.F. működési táblája alapján a Q1 táblából történik.

X	X	X	-
X	X	X	-
X	X	X	-
0	X	X	-
0	0	0	-
0	0	0	-
0	0	0	-
1	0	0	-

$$R1 = \bar{q}_2 \bar{q}_3 \bar{X}_a$$

Z1 kimeneti tábla felvétele az elsődleges ill. a redukált tábla alapján:

1	1	X	-
X	0	0	-
1	1	X	-
X	0	0	-
1	X	X	-
X	0	0	-
0	0	X	-
X	0	0	-

$$Z1 = \bar{q}_1 \bar{q}_2 \bar{q}_3 + \bar{q}_1 q_2 q_3 + q_1 q_2 \bar{q}_3$$

Q2 tábla felvétele:

0	0	0	-
1	0	0	-
1	1	1	-
1	1	1	-
1	1	1	-
0	1	1	-
0	0	0	-
0	0	0	-

$$Q2 = q_2 X_a + \bar{q}_1 q_3 \bar{X}_a + \bar{q}_1 q_2 + q_2 \bar{q}_3$$

S2 a gerjesztési függvényből: $S2 = \bar{q}_1 q_3 \bar{X}_a$

R2 tábla felvétele:

X	X	X	-
0	X	X	-
0	0	0	-
0	0	0	-
0	0	0	-
1	0	0	-
X	X	X	-
X	X	X	-

$$R2 = q_1 q_3 \bar{X} a$$

Z2 kimeneti tábla:

0	0	X	-
X	0	0	-
0	0	X	-
X	0	0	-
0	0	X	-
X	0	0	-
1	1	X	-
X	0	0	-

$$Z2 = q_1 \bar{q}_2 q_3$$

Q3 tábla felvétele:

0	0	1	-
1	1	1	-
1	1	0	-
0	0	0	-
0	0	1	-
1	1	1	-
1	1	0	-
0	0	0	-

$$Q3 = q_3 \bar{X} f + \bar{q}_1 \bar{q}_2 X f + q_1 q_2 X f + \bar{q}_1 \bar{q}_2 q_3 + q_1 q_2 q_3$$

$$S3 \text{ Q3 függvényből kiolvasható: } S3 = \bar{q}_1 \bar{q}_2 X f + q_1 q_2 X f$$

R3 tábla felvétele:

X	X	0	-
0	0	0	-
0	0	1	-
X	X	X	-
X	X	0	-
0	0	0	-
0	0	1	-
X	X	X	-

$$R3 = q_1 \bar{q}_2 X_f + \bar{q}_1 q_2 X_f$$

Z3 kimeneti tábla felvétele:

0	0	X	-
X	1	1	-
0	0	X	-
X	1	1	-
0	0	X	-
X	1	1	-
0	0	X	-
X	1	1	-

$$Z3 = \bar{q}_1 \bar{q}_2 q_3 + \bar{q}_1 q_2 \bar{q}_3 + q_1 q_2 q_3 + q_1 \bar{q}_2 \bar{q}_3$$

2. Feladat

1. ütem: Z1, Z2 tölt Xk-ig
 2. ütem: Z3 tölt Xf-ig
 3. ütem: Z5 ürít Xa-ig (teljesen)
 4. ütem: Z2, Z3 tölt Xf-ig
 5. ütem: Z4, Z5 ürít Xa-ig (teljesen)
- A rendszer kiinduláskor legyen alapállapotban.

1. Elsődleges tábla felvétele:

Állapotkódoláshoz a GRAY-kódot használjuk, a hálózat versenyhelyzet mentes kialakítása érdekében.

Xa, Xk, Xf : bemenetek, Z1, Z2, Z3, Z4, Z5:kimenetek

000	001	011	010	110	111	101	100	Z1	Z2	Z3	Z4	Z5
a	b		-	-		-	-	1	1	0	0	0
	b	c	-	-		-	-	1	1	0	0	0
		c	-	-	d	-	-	0	0	1	0	0
		e	-	-	d	-	-	0	0	0	0	1
	f	e	-	-		-	-	0	0	0	0	1
g	f		-	-		-	-	0	0	0	0	1
g	h		-	-		-	-	0	1	1	0	0
	h	i	-	-		-	-	0	1	1	0	0
		i	-	-	j	-	-	0	1	1	0	0
		k	-	-	j	-	-	0	0	0	1	1
	l	k	-	-		-	-	0	0	0	1	1
a	l		-	-		-	-	0	0	0	1	1

2. Redukált tábla felvétele és az állapot gráf elkészítése:

-ekvivalens állapotok keresésével alakítjuk ki a redukált táblát.

A redukált táblával meg fog egyezni a gerjesztési tábla (k), a vezérlési tábla (k) ill. a kimeneti tábla (k) mérete, peremezése és elrendezése.

000	001	011	010	110	111	101	100
a	b	c	-	-	d	-	-
g	f	l	-	-	d	-	-
g	h	i	-	-	j	-	-
a	l	k	-	-	j	-	-

0	0
0	1
1	1
1	0

A $\log_2 s$ összefüggéssel, ahol s a sorok száma, meghatározzuk a F.F.-ok számát. Mivel a redukált tábla 4 sorból áll így a F.F.-ok száma 2.

2^4	2^3	2^2	2^1	2^0
q ₁	q ₂	Xf	Xk	Xa

A fenti táblázat a redukált tábla, gerjesztési táblák, vezérlési táblák és kimeneti táblák (5 bemeneti változós) peremezését helyiérték szerint mutatja.

3. Gerjesztési tábla felvétele:

00	00	00	-	-	01	-	-
11	01	01	-	-	01	-	-
11	11	11	-	-	10	-	-
00	10	10	-	-	10	-	-

Az RS F.F. vezérlési táblája:

Q_n	Q_{n+1}	R	S	Funkció
0	0	x	0	Tárol
0	1	0	1	Beír
1	0	1	0	Töröl
1	1	0	x	Tiltott

Q1 tábla felvétele:

0	0	0	-	-	0	-	-
1	0	0	-	-	0	-	-
1	1	1	-	-	1	-	-
0	1	1	-	-	1	-	-

$$Q1 = q1q2 + q1Xa + q2\bar{X}k\bar{X}a$$

A gerjesztési függvényből (Q1) a meghúzási kombináció (S1) ki is olvasható, mégpedig az a kombináció amely nem tartalmazza q1-et.

S1 tábla felvétele:

0	0	0	-	-	0	-	-
1	0	0	-	-	0	-	-
X	X	X	-	-	X	-	-
0	X	X	-	-	X	-	-

$$S1 = q2\bar{X}k\bar{X}a$$

R1 tábla felvétele az RS F.F. működési táblája alapján a Q1 táblából.

X	X	X	-	-	X	-	-
0	X	X	-	-	X	-	-
0	0	0	-	-	0	-	-
1	0	0	-	-	0	-	-

$$R1 = \bar{q}2\bar{X}k\bar{X}a$$

Q2 tábla felvétele:

0	0	0	-	-	1	-	-
1	1	1	-	-	1	-	-
1	1	1	-	-	0	-	-
0	0	0	-	-	0	-	-

$$Q2 = \bar{q}1q2 + \bar{q}1Xf + q2\bar{X}f$$

S2 tábla felvétele:

0	0	0	-	-	1	-	-
X	X	X	-	-	X	-	-
X	X	X	-	-	0	-	-
0	0	0	-	-	0	-	-

$$S2 = \bar{q}_1 X f$$

R2 tábla felvétele:

X	X	X	-	-	0	-	-
0	0	0	-	-	0	-	-
0	0	0	-	-	1	-	-
X	X	X	-	-	X	-	-

$$R2 = q_1 X f$$

Z1 kimeneti tábla:

1	1	0	-	-	X	-	-
0	0	0	-	-	0	-	-
0	0	0	-	-	X	-	-
X	0	0	-	-	0	-	-

$$Z1 = \bar{q}_1 \bar{q}_2 \bar{X} k$$

A Z2, Z3, Z4 és Z5 kimeneti függvények kimeneti tábláinak a felvétele és meghatározása, a Z1 kimeneti függvény meghatározásának lépéseivel analóg módon történik.

Az alábbi példaként szerepelő két program, a **tartály töltés/ürítés** és **RS F.F.** megvalósítása mikroprocesszorral, betekintést nyújt a mikroprocesszorok programozásának technikájába, amely a későbbiekben segítséget fog nyújtani a mikroprocesszorok ipari technológiák automatizált működésének a felhasználásához, amelyet a **könyv2a.pdf** file fog tartalmazni.

Feladat: Tartály töltés/ürítés megvalósítása 8086 mikroprocesszorral

Z1 szelep tölt E1-ig, majd Z2 tölt E2-ig ezután Z3 szelepen keresztül leeresztés E0-ig és ezt 13x-szor.

Az INTEL 8255 PIO MODE „0”-ás üzemmódjában a portok bitkiosztását szemlélteti az alábbi részletezés:

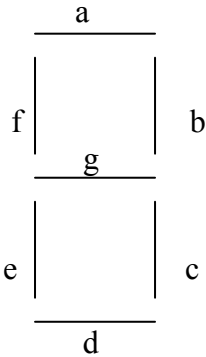
A port (kimenet): 0.,1.,2. biten Z1, Z2, Z3 szelepek

B port (bemenet): 0.,1.,2. biten E0, E1, E2 érzékelők

C port (kimenet): 0.,1.,2.,3.,4.,5.,6. bitek a hétszegmenses kijelző (a,b,c,d,e,f,g) szegmenseit tartalmazza.

Vezérlő byte meghatározása:

				A	C _f		B	C _a	
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀		
1	0	0	0	0	0	1	0	:	82h



t(töltés):

ü(ürítés):

Program:

```

MOV DX, 0303 (PIO konfigurálása)
MOV AL, 82
OUT DX, AL
MOV BL, 0D (13x tölt)
cím 6: MOV AL, 78 (t kijelzése)
MOV DX, 0302
OUT DX, AL
MOV AL, 01 (Z1 nyit)
MOV DX, 0300
OUT DX, AL
MOV DX, 0301
cím: IN AL, DX (?E1)
AND AL, 07
CMP AL, 03
JZ cím1
JMP cím
cím 1: MOV DX, 0300
MOV AL, 02 (Z2 nyit)
OUT DX, AL
MOV DX, 0301
cím 2: IN AL, DX (?E2)
AND AL, 07

```

```

    CMP AL, 07
    JZ cím3
    JMP cím2
cím 3: MOV DX, 0302 (ü kijelzése)
        MOV AL, 1D
        OUT DX, AL
        MOV DX, 0300
        MOV AL, 04 (Z3 nyit)
        OUT DX, AL
        MOV DX, 0301
cím 4: IN AL, DX (?E0)
        AND AL, 07
        CMP AL, 00
        JZ cím5
        JMP cím4
        DEC BL
        JNZ cím6
cím 5: MOV DX, 0302 (0 kijelzése)
        MOV AL, 3F
        OUT DX, AL
        MOV DX, 0300 (szelepek zárása)
        MOV AL, 00
        OUT DX, AL

```

Példa: RS F.F. megvalósítása 8085 mikroprocesszorral

```

    MVI A, 90    A 8255 PIO port kiosztása: A Port Input; B Port Output; C Port közömbös
    OUT C3      A 8255 PIO parancsregiszterének beállítása a fenti port kiosztás szerint
hurok0: IN  A Port címe
        ANI 03
        CPI 00
        JZ hurok
        CPI 01
        JZ hurok1
        CPI 02
        JZ hurok2
        CPI 03
        JZ hurok 0
hurok:  XRA A
        OUT B Port címe
        JMP hurok0
hurok1: MVI A, 01
        OUT B Port címe
        JMP hurok0
hurok2: MVI A, 02
        OUT B Port címe
        JMP hurok0

```

Tárolás: $Q=1$; B Port 0. bit

Törlés: $\bar{Q}=1$; B Port 1. bit

Beírás: $S=1$; $R=0$; A Port 0. bit

Törlés: $R=1$; $S=0$; A Port 1. bit

$S=1$ és az $R=1$ egyidejű vezérlése tiltott kombináció

15. Az alábbiakban közölt feladatsor, az I. minta zárthelyi a III. évfolyam informatikus hallgatói részére

Digitális rendszerek I. Zh.

Informatikus hallgatók részére III. évfolyam

Kidolgozási idő: 60 perc

1. Egyszerűsítse az

$$F(D,C,B,A) = \sum (0,1,2,7,9,11,12,14,15)$$
függvényt diszjunktív és konjunktív alakban!
Realizálja az egyszerűsített függvényeket,
NAND/NAND és NOR/NOR
hálózatokkal!
2. Adja meg az $F^3_{202}(A,B,C)$ függvényt
 - kombinációs táblázatban;
 - KV táblán;
 - teljes diszjunktív normál alakban;
 - teljes konjunktív normál alakban;
 - mintermesen;
 - maxtermesen!
3. Rajzoljon elengedésre és meghúzásra elsőbbséget biztosító relés
alapkapsolásokat (RS0, RS1 relés F.F.-ok)!
4. Készítsen 18-nál leálló, aszinkron bináris előreszámlálót, JK F.F.-ok-
kal!
5. Realizálja az AIKEN/GRAY kódátalakító legmagasabb helyiértékű lo-
gikai függvényét 0-9-ig, n-1 bemenettel (B,C,D) rendelkező adatsze-
lektorral (MUX)!
6. Készítsen 4 bites visszaléptető regisztert D F.F.-okkal!
7. Rajzolja fel a szinkron számlálók sematikus modelljét!
8. Mit ért relatív redundancia alatt és hogyan határozza meg?
9. Adja meg a GRAY 9 és az AIKEN 9 között a Hamming távolságot!
10. Adja meg a legegyszerűbb alakban az $A \& (A+B)$ Boole algebrai
összefüggést!

Digitális rendszerek I. Zh.
Informatikus hallgatók részére III. évfolyam

Kidolgozási idő: 60 perc

1. Minimalizálja az
 $F(D,C,B,A) = \sum (2,3,7,10,11,12,13,14,15) + \sum_x (4,5)$
 függvényt diszjunktív és konjunktív alakban!
 Realizálja az egyszerűsített függvényeket NAND/NAND, NOR/NOR és
 Érintkezős hálózatokkal!
2. Adja meg az $F^3_{193}(A,B,C)$ függvényt
 - kombinációs táblázatban;
 - KV táblán;
 - teljes diszjunktív normál alakban;
 - teljes konjunktív normál alakban;
 - mintermesen;
 - maxtermesen!
3. Mit jelent a TTL NAND elemekkel realizált RS F.F. kimenetén, ha a bemeneten S=0 és R=1 vezérlést kap?
4. Készítsen 17-ig ismétlő, aszinkron bináris előreszámlálót JK F.F.-okkal!
5. Realizálja az JOHNSON/GRAY kódátalakító legmagasabb helyiértékű gikai függvényét 0-7-ig, adatszelektorral (MUX)!
6. Készítsen 4 bites előreléptető regisztert D F.F.-okkal!
7. Realizálja a teljes összeadó logikai függvényeit DC áramkörrel!
8. Adja meg annak az aszinkron sorrendi áramkör RS tárolójának az S vezérlési függvényét, amely a következő ütem szerint működik:
 1. ü. t. x_k -ig;
 2. ü. t. x_r -ig;
 3. ü. ü. x_a -ig!
9. Mit ért hazárdon és versenyhelyzet alatt (mintapéldák)?
10. Helyes-e az alábbi gerjesztési függvény

$$Q = q \bar{x}_1 + x_1 \bar{x}_2 ?$$

Megjegyzés:

A kombinációs és a sorrendi hálózatok alábbi „.sch” kiterjesztésű file-jai, a **Tina Pro** áramkör szerkesztő program installálását követően működőképesen láthatók, az **Analízis/Digitális nyomkövetés** menü pontjában.

16. Kidolgozott, szimulációra alkalmas hálózatok

1. Igen.sch
2. Nem.sch
3. And.sch
4. Or.sch
5. Nand.sch
6. Nor.sch
7. Exor.sch
8. Megadao.sch
9. Megadoa.sch
10. Egyszao.sch
11. Egyszoa.sch
12. Kódátal.sch
13. Dec.sch
14. Mux.sch
15. Rsff.sch
16. Jkff.sch
17. Dff.sch
18. MSff.sch
19. abr1.sch: 4 bites jobbra léptető regiszter D F.F.-okkal felépítve
20. abr2.sch: 4 bites jobbra léptető regiszter JK F.F.-okkal kialakítva
21. abr3.sch: 4 bites balra léptető regiszter D FF.-ok alkalmazásával
22. abr4.sch: 4 bites bináris aszinkron előreszámláló JK F.F.-okból felépítve
23. abr5.sch: 4 bites bináris aszinkron visszaszámláló JK F.F.-ból
24. abr6.sch: 4 bites bináris aszinkron ismétlő számláló kialakítása JK F.F.-okkal
25. abr7.sch: 4 bites bináris aszinkron leálló számláló JK F.F.-okkal
26. abr8.sch: JK F.F.-okból kialakított, aszinkron számlálóval megvalósított fűtőelem ki/bekapcsolása (a helyes működés idődiagramon is nyomon követhető)
27. abr9.sch: 4 bites JOHNSON szinkron előreszámláló JK F.F.-okkal megvalósítva
28. abr10.sch: 4 bites JOHNSON szinkron előreszámláló D F.F.-okkal kialakítva (a helyes működést idődiagramon is láthatjuk)
29. abr11.sch 4 bites JOHNSON szinkron visszaszámláló JK FF.-okkal felépítve (a helyes működés idődiagramon is látható)
30. abr12.sch 4 bites JOHNSON szinkron visszaszámláló D FF.-okból készítve (az ábra idődiagram megjelenítésére is alkalmas)

A fentiekben említett „.sch” kiterjesztésű file-okat a „**Hálózatok**” című, valamint a szimulációhoz szükséges szoftvereket install formában a „**Tina_Pro_5.1**” és a „**Logikai Szimulátor**” könyvtárak tartalmazzák.

Mind az oktatási anyag, mind pedig a szimulációk használata lehetőséget ad a hallgatóknak otthoni körülmények között, az un. „**Virtuális Laboratórium**” megteremtéséhez.

Dr. Gárdus Zoltán Ph.D.
egyetemi adjunktus
Villamosmérnöki Intézet
Automatizálási Tanszék

Irodalomjegyzék

- [1] Ajtonyi István: DIGITÁLIS RENDSZEREK
MISKOLCI EGYETEM, 2002
- [2] Ajtonyi István: Vezérléstechnika I.-II.
Tankönyvkiadó, Budapest, 1991
- [3] Bánhidi László-Oláh Miklós: AUTOMATIKA mérnököknek
Gyuricza István-Kiss Mátyás
Rátkai László- Szecső Gusztáv
Tankönyvkiadó, Budapest, 1992
- [4] Gál T: Programozható logikák BME
Tankönyvkiadó, 1994
- [5] INTEL SDK-85 System Design Kit User's Manual
INTEL Corporation, 1978
- [6] INTEL SDK-86 System Design Kit User's Manual
INTEL Corporation, 1978
- [7] Tina_Pro_5,1: Áramkör szerkesztő szoftver
The Complete Electronics Lab
- [8] XILINX The Programmable Logic Data Book San Jose, California 95124, 1994
- [9] László József: A PC hardver programozása
Valós és védett módban
PASCAL és ASSEMBLY nyelven
ComputerBooks
Budapest, 2003