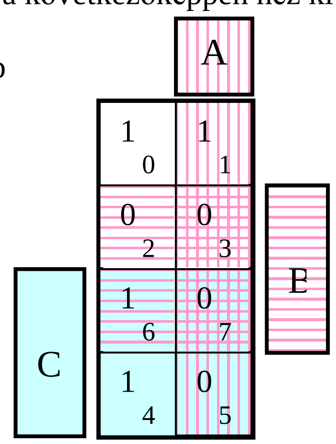
Vegyük a következő igazságtáblával rendelkező F logikai függvényt:

| n | C | B | A | F | $\overline{F}$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 | 1 |
| 4 | 1 | 0 | 0 | 1 | 0 |
| 5 | 1 | 0 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 | 1 | 0 |
| 7 | 1 | 1 | 1 | 0 | 1 |

| mintermek | maxtermek |
|---|---|
| $\overline{C}\cdot\overline{B}\cdot\overline{A}=1$ | $C+B+A=0$ |
| $\overline{C}\cdot\overline{B}\cdot A=1$ | $C+B+\overline{A}=0$ |
| $\overline{C}\cdot B\cdot\overline{A}=1$ | $C+\overline{B}+A=0$ |
| $\overline{C}\cdot B\cdot A=1$ | $C+\overline{B}+\overline{A}=0$ |
| $C\cdot\overline{B}\cdot\overline{A}=1$ | $\overline{C}+B+A=0$ |
| $C\cdot\overline{B}\cdot A=1$ | $\overline{C}+B+\overline{A}=0$ |
| $C\cdot B\cdot\overline{A}=1$ | $\overline{C}+\overline{B}+A=0$ |
| $C\cdot B\cdot A=1$ | $\overline{C}+\overline{B}+\overline{A}=0$ |

Az n a CBA által meghatározott bináris szám decimális értéke (A legmagasabb helyiérték C, a legalacsonyabb A; n = $A2^0 + B2^1 + C2^2 + \ldots$)
A CBA változókkal a hetedik oszlopban kiírt logikai kapcsolatokat megvizsgálva megállapíthatjuk, hogy mindegyik csak akkor ad logikai „1”-t, ha az ö sorának megfelelő CBA értékvariáció áll elő, egyébként logikai „0”. Ezért ezeket a három vátozó elemi logikai AND függvényeinek, másnéven AND-termeknek vagy **minterm**eknek nevezik.
A termek segítségével előállítható F függvény is:
$F=\overline{C}\cdot\overline{B}\cdot\overline{A}+\overline{C}\cdot\overline{B}\cdot A+C\cdot\overline{B}\cdot\overline{A}+C\cdot B\cdot\overline{A}$ Ez az ún **kanonikus diszjunktív alak**. Ebben az alakban az egyes **mintermek OR kapcsolatá**val adjuk meg a logikai függvényt.
A termeket felhasználva tetszőleges logikai függvény előállítható a logikai „AND”, „OR”, „NOT” művelet segírségével, így elvileg a fenti műveleteket elvégző áramkörökkel a függvény realizálható.
F realizálásához így (a NOT műveleteket nem számolva) 8db AND és 3db OR (összesen 11) műveletet kell elvégezni. Lehet ennél kevesebb is? Egyszerűsítsünk.
Algerailag:
$F=\overline{C}\cdot\overline{B}\cdot\overline{A}+\overline{C}\cdot\overline{B}\cdot A+C\cdot\overline{B}\cdot\overline{A}+C\cdot B\cdot\overline{A}=\overline{C}\cdot\overline{B}\cdot\left(\overline{A}+A\right)+C\cdot\overline{A}\cdot\left(\overline{B}+B\right)=\overline{C}\cdot\overline{B}+C\cdot\overline{A}$
Ez az ún. **legegyszerűbb kanonikus diszjunktív alak**.
Így már csak 2db AND és 1db OR műveletet kell elvégezni.
Most még az algebrai módszer egyszerű volt, de bonyolultabb függvényeknél nem biztos, hogy egyértelműen látszik, mit mivel kell összevonni (az előző példában is lehet úgy összvonni, hogy az eredmény bonyolultabb legyen). Ennek megkönnyítésére alkalmazzák a Karanaugh táblás egyserűsítést. Az F függvény Karanaugh táblája a következőképpen néz ki:




A Karanaugh táblában minden egyes termnek (minden CBA variációnak) megfelel egy cella. Az A=1 nek az A-val jelölt oszlop felel meg, a B=1-nek a B-vel jelölt két sor, C=1-nek a C-vel jelölt két sor felel meg (ezt most könnyítésként a cellák háttere is jelzi). A jobb alsó sarokban a CBA variációnak megfelelő n érték adja a cella sorszámát. A cellák úgy vannak elrendezve, hogy függőlegesen, vagy vízszintesen szomszédos cellába csak egy változó megváltoztatásával kelljen lépni. Ez igaz akkor is, ha a változókat nem a feltüntetett sorrendben vesszük fel, csak ekkor a cellák sorszámozása megváltozik.
Az n sorszámú cellába „1”-t írunk, ha F értéke az n. CBA variáció esetén 1, egyébként „0”-t.
Ha függőlegesen, vagy vízszintesen szomszédos cellában „1”-t találunk, akkor tudhatjuk, hogy az egyik és csak az egyik változó megváltozása ellenére az „1” maradt, így a két szomszédos „1” attól a változótól függetlenül áll elő. Pl. a 6. és a 4. cellában mindkét esetben



PDF created with FinePrint pdfFactory trial version http://www.fineprint.com

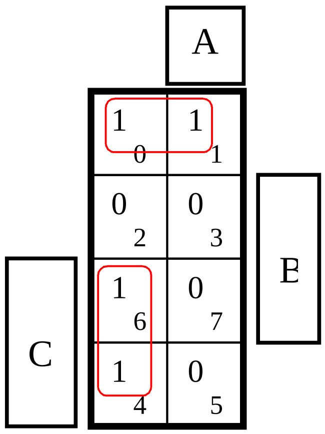
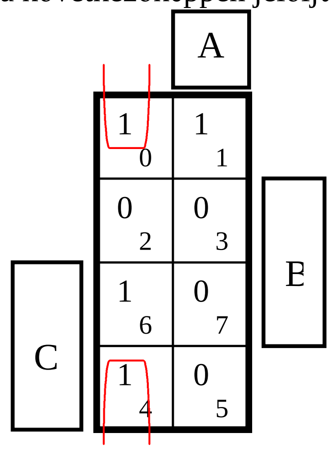
A= „0” és C= „1”, de B a két cellában nem egyezik meg. Így, ha a 6. és a 4. cellát egy egységként kezeljük, akkor az független B-től. Tehát a 6. és a 4. cellára:

$F_{6,4} = C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot \overline{A} \Rightarrow F_{6,4} = C \cdot \overline{A}$

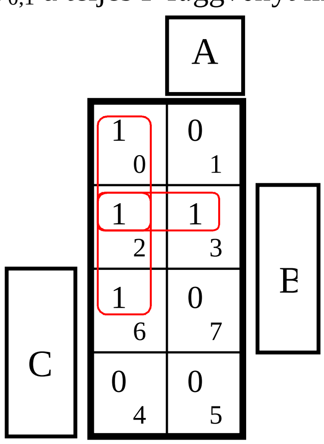
Ugyanígy a 0. és 1. cellákra:

$F_{0,1} = \overline{C} \cdot \overline{B} \cdot \overline{A} + \overline{C} \cdot \overline{B} \cdot A \Rightarrow F_{0,1} = \overline{C} \cdot \overline{B}$

Itt már nem algebrai egyszerűsítéseket kellett keresni, hanem egy ábrán szomszédos elemeket, ami lényegesen áttekinthetőbb. Az összevonást a következőképpen jelöljük:

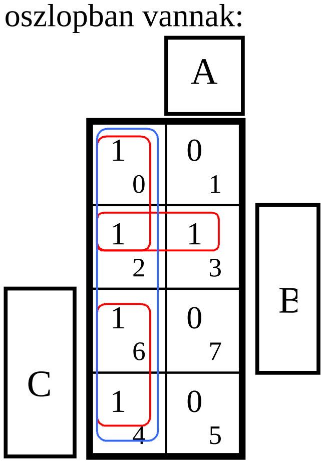



A Karnaugh táblában a tábla párhuzamos szélei mentén lévő elemek is szomszédosnak tekinthetők (mintha kivágva összeragasztanánk hengerré vagy a függőleges, vagy a vízszintes élei mentén), így a fenti ábrán látható összevonás is lehetséges. Más kérdés, hogy most semmi szükség rá, hiszen az eredeti két összevonással minden „1” értékű cellát megadtunk ($F_{6,4}$ és $F_{0,1}$ a teljes F függvényt megadja.) Néhány lehetséges összevonás:



Itt $F = \overline{C} \cdot \overline{B} \cdot \overline{A} + \overline{C} \cdot B \cdot \overline{A} + \overline{C} \cdot B \cdot A + C \cdot B \cdot \overline{A}$ (11 művelet)
Itt a 2 termet mindhárom összevonáshoz fel kellett használnunk.
Egyszerűsítés után:
$F = \overline{C} \cdot \overline{A} + \overline{C} \cdot B + B \cdot \overline{A}$ (5 művelet) (**legegyszerűbb kanonikus diszjunktív alak**)
Ha egy termet többszörösen használunk fel, a függvény tovább egyszerűsíthető. Itt:
$F = \overline{C} \cdot \left(\overline{A} + B\right) + B \cdot \overline{A}$ (4 művelet) (**minimál alak**)
Tehát a minimál alak és a legegyszerűbb kanonikus diszjunktív alak nem minden eset egyezik meg. (Mert az utóbbit köti a diszjunktív alak formai előírása: mintermek OR kapcsolata)

A Karnaugh táblán a kettős összevonások is összevonhatók, ha egy sorban vagy egy oszlopban vannak:



$$F = \overline{C} \cdot \overline{B} \cdot \overline{A} + \overline{C} \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot \overline{A} + $$
$$+ \overline{C} \cdot B \cdot A + C \cdot B \cdot \overline{A}$$

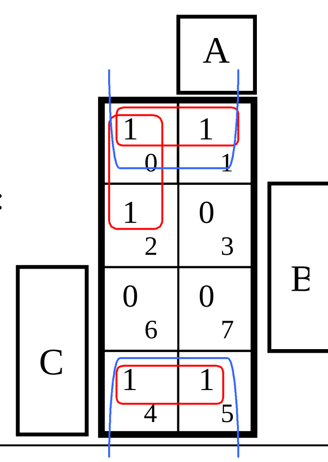
Ebből:

$F = \overline{A} + \overline{C} \cdot B$

A jobb oldali:

$$F = \overline{C} \cdot \overline{B} \cdot \overline{A} + \overline{C} \cdot \overline{B} \cdot A + \overline{C} \cdot B \cdot \overline{A} + $$
$$+ C \cdot \overline{B} \cdot \overline{A} + C \cdot \overline{B} \cdot A$$

$F = \overline{B} + \overline{C} \cdot \overline{A}$





PDF created with FinePrint pdfFactory trial version http://www.fineprint.com

Az egyszerűsítéseket sokszor nem diszjunkt alakban könnyebb elvégezni. A másik lehetőség az ún. kanonikus konjunktív alak. Ezt a legelső példafüggvényre megkaphatjuk, ha a F negáltjára írjuk fel a kanonikus diszjunkt alakot:

$\overline{F} = \overline{C} \cdot B \cdot \overline{A} + \overline{C} \cdot B \cdot A + C \cdot \overline{B} \cdot A + C \cdot B \cdot A$

Ebből:

$$F = \overline{\overline{F}} = \left(\overline{\overline{C} \cdot B \cdot \overline{A} + \overline{C} \cdot B \cdot A + C \cdot \overline{B} \cdot A + C \cdot B \cdot A}\right)$$

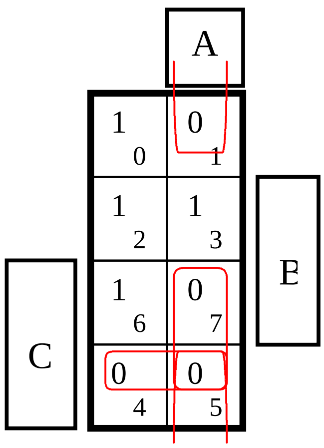
Felhasználva a De-Morgan azonosságokat:

$$F = \overline{\overline{F}} = \left(\overline{\overline{C} \cdot B \cdot \overline{A}}\right) \cdot \left(\overline{\overline{C} \cdot B \cdot A}\right) \cdot \left(\overline{C \cdot \overline{B} \cdot A}\right) \cdot \left(\overline{C \cdot B \cdot A}\right) =$$

$$= \left(C + \overline{B} + A\right) \cdot \left(C + \overline{B} + \overline{A}\right) \cdot \left(\overline{C} + B + \overline{A}\right) \cdot \left(\overline{C} + \overline{B} + \overline{A}\right)$$ Ez a **kanonikus konjunktív alak**.

Az ebben az alakban szerplő elemi logikai függvényeket a három vátozó elemi logikai OR függvényeinek, vagy OR-termnek, vagy maxtermnek hívják. Ezeknek jellegzetessége, hogy a logikai “0”-értéket csak egyetlen CBA variáció esetén adnak, egybként értékük “1”. A táblázatban fel van tüntetve, hogy melyik maxterm melyik CBA variációra állítja elő a logokai ”0”-t.

A kanonikus konjunktív alakot is lehet egyszerűsíteni. Annyiban különbözik a kanonikus diszjunkt alak egyszerűsítésétől, hogy a Karnaugh táblában nem a szomszédos “1”-ket, hanem a szomszédos “0”-kat kell összevonni a fenn megismert módon. Ezzel előállítottuk $\overline{F}$ legegyszerűbb diszjunkt alakját. Ezt negálva megkapjuk F-t, a diszjunkt alakból a De-Morgan azonosságokat felhasználva konjunktív alak lesz. Pl.:




Itt $F = \overline{C} \cdot \overline{B} \cdot \overline{A} + \overline{C} \cdot B \cdot \overline{A} + \overline{C} \cdot B \cdot A + C \cdot B \cdot \overline{A}$

Itt a 2 termet mindhárom összevonáshoz fel kellett használnunk.

Egyszerűsítés után:

$\overline{F} = C \cdot A + C \cdot \overline{B} + \overline{B} \cdot A$

$F = \left(\overline{C \cdot A + C \cdot \overline{B} + \overline{B} \cdot A}\right) = \left(\overline{C \cdot A}\right) \cdot \left(\overline{C \cdot \overline{B}}\right) \cdot \left(\overline{\overline{B} \cdot A}\right) =$ (5 művelet)

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

(**legegyszerűbb kanonikus konjunktív alak**)

A zárójeleket felbontva és összegezve:

$F = \overline{C} \cdot \left(\overline{A} + B\right) + B \cdot \overline{A}$ (4 művelet) (**minimál alak**)

Tehát a minimál alak és a legegyszerűbb kanonikus konjunktív alak sem minden eset egyezik meg. (Az utóbbit is köti a konjunktív alak formai előírása: maxtermek AND kapcsolata)

Látható, hogy a minimál alak mind a diszjunktív, mind a konjunktív alak egyszerűsítésével kiszámítható.

Összefoglalva:

> A kanonikus diszjunktív alak azoknak a mintermeknek OR kapcsolatát adja meg, melyeknél F = „1”.
>
> A kanonikus konjunktív alak azoknak a maxtermeknek AND kapcsolatát adja meg, melyeknél F = „0”.

Egy N változós logikai függvényeknél a lehetséges mintermek és maxtermek száma $2^N$. Ha az eddig ismertetett módon írnánk fel a függvényt, azaz minden egyes mintermet, vagy maxtermet teljesen kiírnánk, akkor a függvény meglehetősen hosszú és átláthatatlan módon lenne megadva. Ezért az egyes mintermeket ($m_n$) és maxtermeket ($M_n$) megsorszámozzák. A sorszám megállapításához a termben lévő változó értékét „0”-nak vesszük, ha az negálva van, illetve „1”-nek vesszük ha nincs negálva. Ebből a sorszám (ha A változó a legkisebb helyiérték):

$s = A \cdot 2^0 + B \cdot 2^1 + C \cdot 2^2 + \ldots + (utolsó\_változó) \cdot 2^{N-1}$



PDF created with FinePrint pdfFactory trial version http://www.fineprint.com

Pl: $\overline{C}\cdot B\cdot \overline{A} \Rightarrow s = 0\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0 = 2 \Rightarrow \overline{C}\cdot B\cdot \overline{A} = m_2$

$C\cdot \overline{B}\cdot A \Rightarrow s = 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 = 5 \Rightarrow C\cdot \overline{B}\cdot A = m_5$

És: $C + \overline{B} + \overline{A} \Rightarrow s = 1\cdot 2^2 + 0\cdot 2^1 + 0\cdot 2^0 = 4 \Rightarrow C + \overline{B} + \overline{A} = M_4$

$\overline{C} + B + A \Rightarrow s = 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 3 \Rightarrow \overline{C} + B + A = M_3$

Ennek segítségével egy logikai függvény megadásához elég a megfelő termeket felsorolni.
Pl. az első példa kanonikus diszjunktív alakja: $F^3 = F(CBA) = \sum_3 (0,1,4,6)$

És kanonikus konjunktív alakja: $F^3 = F(CBA) = \prod_3 (0,2,4,5)$

A konjunktív alakban azoknak a mintermeknek megfelelő maxtermek vannak felsorolva, amelyek a diszjunktív alakból kimaradtak. Ezeknek a sorszáma 2, 3, 5, 7; ami nem egyezik meg a maxterm sorszámával. Ez a táblázatban is látható, az n = 5 –höz tartozó sorban az $m_5$ mintrem és az $M_2$ maxterm van, továbbá elmondhatjuk, hogy $\overline{m_5} = M_2$, mivel az egyik '1'-t a másik '0'-t ad ha az n=5 sorhoz tartozó bemeneti kombináció előáll.
Általánosan ha N változós függvényünk van és a minterm sorszáma sm, akkor a vele egy sorba elhelyeszkedő maxterm sorszáma:

$sM = 2^N - 1 - sm$

és ugyanez igaz fordítva:

$sm = 2^N - 1 - sM$

Általánosabban, figyelembe véve a termek értékét:

$\overline{m_{sm}} = M_{2^N - 1 - sm}$

$\overline{M_{sM}} = m_{2^N - 1 - sM}$

A sorszámváltást az magyarázza, hogy az igazságtáblázatban, ahol a mintermek és a maxtermek fel vannak sorolva, a mintermek sorszámozása a táblázat tetejétől lefelé, a maxtermeké az táblázat aljától felfelé történik.
Ez a sorszámátváltás akkor lényeges, amikor a Karnaugh táblába kell bejelölni az egyes termeket. A minterm sorszáma megegyezik a Karnaugh tábla cellájának sorszámával (n=sm), így azokba a cellákba kell „1"-ket írni, amelyek a diszjunktív alakban meg lettek adva. A konjunktív alakban megadott logikai függvény „0"-inak helyét azonban az átváltás felhasználásával lehet megadni: $n = sm = 2^N - 1 - sM$

Pl. $F^3 = F(CBA) = \prod_3 (0,2,4,5)$

$sM = 0 \Rightarrow sm = 2^3 - 1 - 0 = 7$

$sM = 2 \Rightarrow sm = 2^3 - 1 - 2 = 5$

$sM = 4 \Rightarrow sm = 2^3 - 1 - 4 = 3$

$sM = 2 \Rightarrow sm = 2^3 - 1 - 2 = 5$

Tehát a Karnaugh tábla 2, 3, 5, 7 sorszámú celláiba kerül „0", a többibe „1"

| | | A | |
|---|---|---|---|
| | 1 (0) | 1 (1) | |
| | 0 (2) | 0 (3) | B |
| C | 1 (6) | 0 (7) | B |
| C | 1 (4) | 0 (5) | |

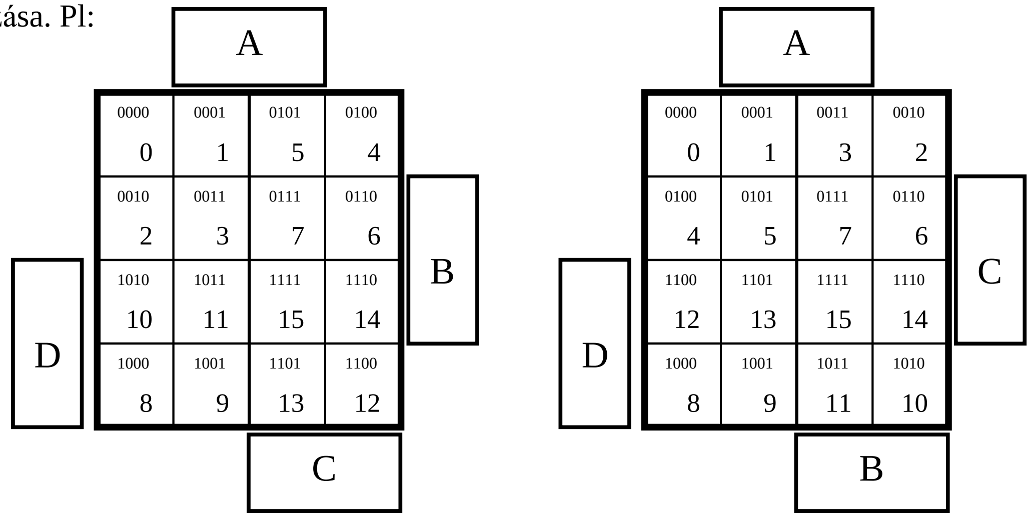


PDF created with FinePrint pdfFactory trial version http://www.fineprint.com

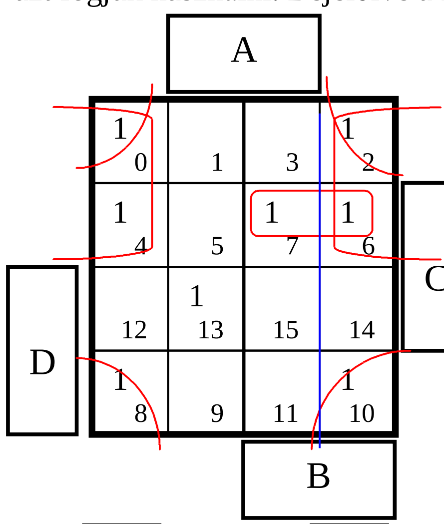
Gyakorlás:
Nézzük a következő négyváltozós föggvényt:

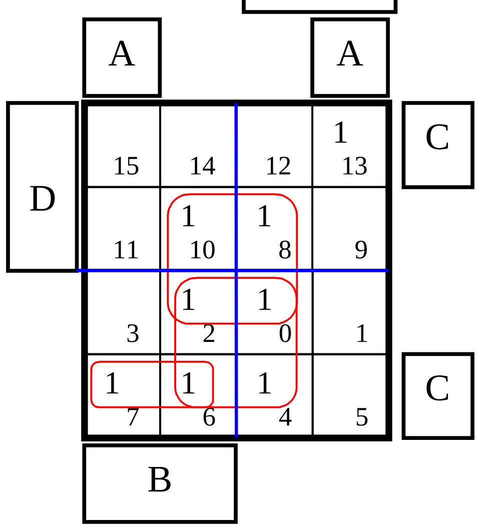
$$F^4 = F(DCBA) = \sum_4 (0,2,4,6,7,8,10,13)$$

Ezt négyváltozós Karanaugh táblában tudjuk ábrázolni. A négyváltozós Karnaugh tábla peremezését (Az A, B, C, D sorok/oszlopok kijelölését) tetszőleges sorrendben vehetjük fel, csak figyelembe kell venni, hogy ha megváltoztatuk a peremezést, megváltozik a cellák sorszámozása. Pl:



Bármelyik változat használható, de az utóbbiban könnyebb a cellák sorszámát fölvenni, ezért azt fogjuk használni. Bejelölve a függvény mintermjeit és egyszerűsítve:





A 13-as mintermnek nincsenek szomszédai, ezért nem egyszerűsíthető, függ mind a 4 változótól: $D \cdot C \cdot \overline{B} \cdot A$
$D = 1 \Rightarrow D; C = 1 \Rightarrow C; B = 0 \Rightarrow \overline{B}; A = 1 \Rightarrow A$.
A 7-es és a 6-os mintermeket összevontuk. D minkét mintermre „0”, C mindkét mintermre „1”, B mindkét mintermre „1”, A az egyikre „1” a másikra „0”. Tehát ha a DCB=011 bemeneti variáció áll elő, F függetlenül A értékétől „1”-t ad. Ezért ezt az összevonást a $\overline{D} \cdot C \cdot B$ elemi függvény írja le.
Egy összevonás azoktól a változóktól független, amelyeknek a “határát” az átlépi. A 6-7 összevonás pl. A határát lépi át (kék vonal).
A 0-2-4-6 összevonás C és B határát lépi át, ezért attól független, így ezt a $\overline{D} \cdot \overline{A}$ elemi függvény írja le.
A 0-2-8-10 összevonás D és B határait lépi át (látható az árendezett ábrán, kék vonalak), ezért a $\overline{C} \cdot \overline{A}$ elemi függvény írja le.
Az F függvény legegyszerűbb diszjunkt alakja ezek alapján:

$$F = D \cdot C \cdot \overline{B} \cdot A + \overline{D} \cdot C \cdot B + \overline{D} \cdot \overline{A} + \overline{C} \cdot \overline{A}$$

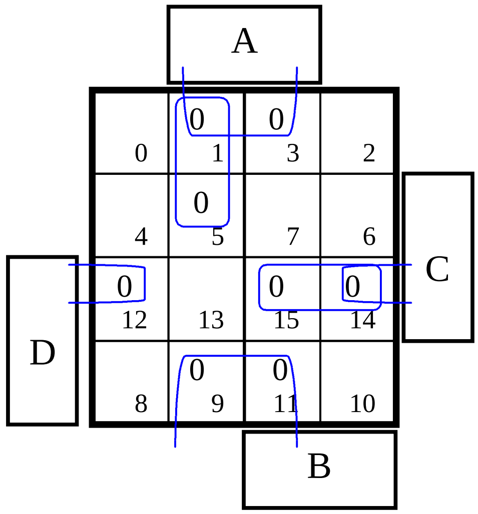
Ez tovább egyszerűsíthető:

$$F = C \cdot \left(D \cdot \overline{B} \cdot A + \overline{D} \cdot B\right) + \left(\overline{D} + \overline{C}\right) \cdot \overline{A}$$

A konjunktív alak a Karnaugh tábla “0” bejegyzései alapján írható fel.
“0” bejegyzést a következő mintermek tartalmaznak:
1, 3, 5, 9, 11, 12, 14, 15



PDF created with FinePrint pdfFactory trial version http://www.fineprint.com




A $sM = 2^N - 1 - sm$ összefüggést felhasználva (N = 4) ezek a következő maxtermeknek felelnek meg:
14, 12, 10, 6, 4, 3, 1, 0
Ebből a konjunktív alak:
$$F^4 = F(DCBA) = \prod_4 (0,1,3,4,6,10,12,14)$$
A konjunktív alak egyszerűsítését a Karnaugh tábla „0” bejegyzéseinek összevonásaival érhetjük el. Így felírható a negáltfüggvény legegyszerűbb diszjunkt alakja a korábban bemutatott módon:
$$\overline{F} = \overline{D} \cdot \overline{B} \cdot A + D \cdot C \cdot \overline{A} + D \cdot C \cdot B + \overline{C} \cdot A$$
Negálással előállítva F-t, felhasználva a De-Morgan azonossságokat $\left( \begin{array}{l} \overline{A + B} = \overline{A} \cdot \overline{B}; \\ \overline{A \cdot B} = \overline{A} + \overline{B} \end{array} \right)$:
$$F = \overline{\overline{F}} = \overline{\overline{D} \cdot \overline{B} \cdot A + D \cdot C \cdot \overline{A} + D \cdot C \cdot \overline{B} + \overline{C} \cdot A} = \left(\overline{\overline{D} \cdot \overline{B} \cdot A}\right) \cdot \left(\overline{D \cdot C \cdot \overline{A}}\right) \cdot \left(\overline{D \cdot C \cdot B}\right) \cdot \left(\overline{\overline{C} \cdot A}\right) =$$
$$= \left(D + B + \overline{A}\right) \cdot \left(\overline{D} + \overline{C} + A\right) \cdot \left(\overline{D} + \overline{C} + \overline{B}\right) \cdot \left(\overline{A} + C\right)$$
Ez a legegyszerűbb konjunktív alak.
Minimalizáláshoz végezzük el a műveleteket és használjuk ki, hogy a következő alakok között lehet egyszerűsíteni:
$$X \cdot Y + X \cdot Y = X \cdot Y$$
$$X \cdot Y \cdot \overline{X} = 0$$

$$F = \left(D + B + \overline{A}\right) \cdot \left(\overline{D} + \overline{C} + A\right) \cdot \left(\overline{D} + \overline{C} + \overline{B}\right) \cdot \left(\overline{A} + C\right) =$$
$$= \left(D\overline{D} + B\overline{D} + \overline{A} \cdot \overline{D} + D\overline{C} + B\overline{C} + \overline{A} \cdot \overline{C} + DA + BA + \overline{A}A\right) \cdot \left(\overline{D} + \overline{C} + \overline{B}\right) \cdot \left(\overline{A} + C\right) =$$
$$= \left(B\overline{D} + \overline{A} \cdot \overline{D} + D\overline{C} + B\overline{C} + \overline{A} \cdot \overline{C} + DA + BA\right) \cdot \left(\overline{D} + \overline{C} + \overline{B}\right) \cdot \left(\overline{A} + C\right)$$

$$F = \left(B\overline{D} + \overline{A} \cdot \overline{D} + D\overline{C} + B\overline{C} + \overline{A} \cdot \overline{C} + DA + BA\right) \cdot \left(\overline{D} + \overline{C} + \overline{B}\right) \cdot \left(\overline{A} + C\right) =$$
$$= \left( \begin{array}{l} B\overline{D} \cdot \overline{D} + \overline{A} \cdot \overline{D} \cdot \overline{D} + D\overline{C} \cdot \overline{D} + B\overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{C} \cdot \overline{D} + DA\overline{D} + BA\overline{D} + \\ + B\overline{D} \cdot \overline{C} + \overline{A} \cdot \overline{D} \cdot \overline{C} + D\overline{C} \cdot \overline{C} + B\overline{C} \cdot \overline{C} + \overline{A} \cdot \overline{C} \cdot \overline{C} + DA\overline{C} + BA\overline{C} + \\ + B\overline{D} \cdot \overline{B} + \overline{A} \cdot \overline{D} \cdot \overline{B} + D\overline{C} \cdot \overline{B} + B\overline{C} \cdot \overline{B} + \overline{A} \cdot \overline{C} \cdot \overline{B} + DA\overline{B} + BA\overline{B} \end{array} \right) \cdot \left(\overline{A} + C\right) =$$
$$= \left( \begin{array}{l} B\overline{D} + \overline{A} \cdot \overline{D} + B\overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{C} \cdot \overline{D} + BA\overline{D} + \\ + D\overline{C} + B\overline{C} + \overline{A} \cdot \overline{C} + DA\overline{C} + BA\overline{C} + \\ + \overline{A} \cdot \overline{D} \cdot \overline{B} + D\overline{C} \cdot \overline{B} + \overline{A} \cdot \overline{C} \cdot \overline{B} + DA\overline{B} \end{array} \right) \cdot \left(\overline{A} + C\right)$$

Kihasználjuk, hogy
$$X \cdot Y \cdot Z + X \cdot Y = X \cdot Y \cdot (Z + 1) = X \cdot Y$$



PDF created with FinePrint pdfFactory trial version http://www.fineprint.com

$$F=\left(B\overline{D}+\overline{A}\cdot\overline{D}+D\overline{C}+\overline{A}\cdot\overline{C}+B\overline{C}+AB\overline{D}+A\overline{B}D\right)\cdot\left(\overline{A}+C\right)=$$

$$=\begin{pmatrix}B\overline{D}\cdot\overline{A}+\overline{A}\cdot\overline{D}\cdot\overline{A}+D\overline{C}\cdot\overline{A}+\overline{A}\cdot\overline{C}\cdot\overline{A}+B\overline{C}\cdot\overline{A}+AB\overline{D}\cdot\overline{A}+A\overline{B}D\overline{A}+\\+B\overline{D}C+\overline{A}\cdot\overline{D}C+D\overline{C}C+\overline{A}\cdot\overline{C}C+B\overline{C}C+AB\overline{D}C+A\overline{B}DC\end{pmatrix}=$$

$$=\begin{pmatrix}B\overline{D}\cdot\overline{A}+\overline{A}\cdot\overline{D}+D\overline{C}\cdot\overline{A}+\overline{A}\cdot\overline{C}+B\overline{C}\cdot\overline{A}+\\+B\overline{D}C+\overline{A}\cdot\overline{D}C+AB\overline{D}C+A\overline{B}DC\end{pmatrix}=$$

$$=\left(\overline{A}\cdot\overline{D}+\overline{A}\cdot\overline{C}+B\overline{D}C+A\overline{B}DC\right)=\overline{A}\cdot\left(\overline{D}+\overline{C}\right)+C\left(B\overline{D}+A\overline{B}D\right)$$

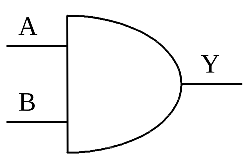
Ugyanazt a minimálalakot kaptuk meg, mint a diszjunkt alaknál, csak lényegesen több számolással.

A függvényt megvalósító áramkört, (ha más, pl. hazárd kritérium nincs) célszerű a minimálalak alapján megszerkeszteni, mert ez igényli a legkevesebb kapuáramkört. Ha diszjunktív vagy konjunktív alakból indulunk ki, akkor AND, OR, és NOT műveletekből rakhatjuk össze a logikai függvényt.

Alalpkapuk:

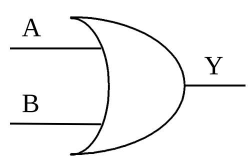
AND:

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



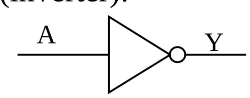

OR:

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |




NOT (inverter):

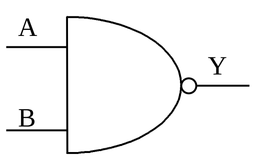
| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |




További lehetőség az AND, és az OR művelet a NOT művelettel összevonása:

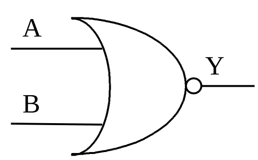
NAND (NOT-AND):

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |




NOR (NOT-OR):

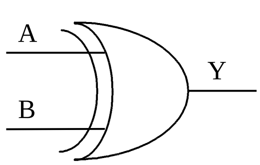
| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |




Bizonyos esetekben hasznos lehet az XOR művelet is:

XOR:

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |




Ha csak kevés számú XOR kapu kell, azt megvalósíthatjuk az $F=A\overline{B}+\overline{A}B$ függvény realizálásával is.

A realizálás lépéseit nézzük meg a példafüggvény minimális alakján:

$$F=C\cdot\left(D\cdot\overline{B}\cdot A+\overline{\overline{D}\cdot B}\right)+\left(\overline{D}+\overline{C}\right)\cdot\overline{A}$$

Két fő lépés kell megtenni:

- Alulról felefelé haladva meg kell határozni a logikai függvény szinjeit
- Felülről lefelé szintenként realizálni a függvényt



PDF created with FinePrint pdfFactory trial version http://www.fineprint.com

A szintek meghatározása:

Azt a logikai műveletet keressük, amit már nem lehet tovább bontani, a legkisebb logikai egységek között teremt kapcsolatot. Egy logikai szintnek számít az a logikai részfüggvény, melyben a logikai egységek (változók és/vagy részfüggvények) között azonos logikai művelet van.

*Minta*: $C \cdot \left(ABCD + A\overline{D} + \overline{B} + \overline{C}\right)$.

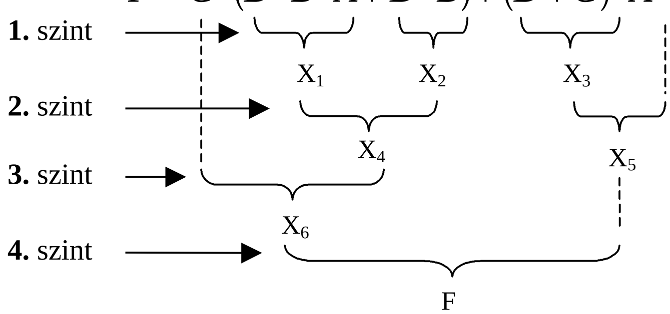
Nem lehet tovább bontani a ABCD AND műveletének operandusait, $A\overline{D}$ AND műveletének operandusait. Az első szintet ez a két AND művelet által megvalósított részfüggvény alkotja.

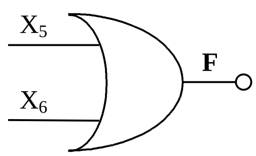
Ha az első szint részfüggvényeit önálló változóként kezeljük, nem lehet tovább bontani a zárójel OR műveletének operandusait (ABCD, $A\overline{D}$ részfüggvények; $\overline{B}$, $\overline{C}$ változók). A második szintet az OR művelet által megvalósított részfüggvény alkotja.

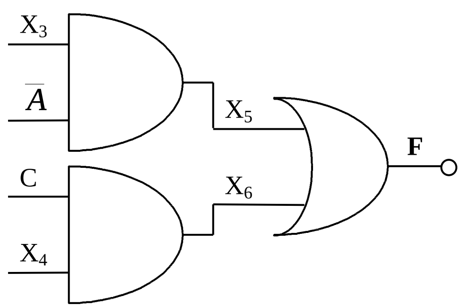
Ha a második szint részfüggvényét önálló változóként kezeljük, nem lehet tovább bontani a C és a zárójeles művelet közötti AND műveletet, ez a legfelső harmadik szint.

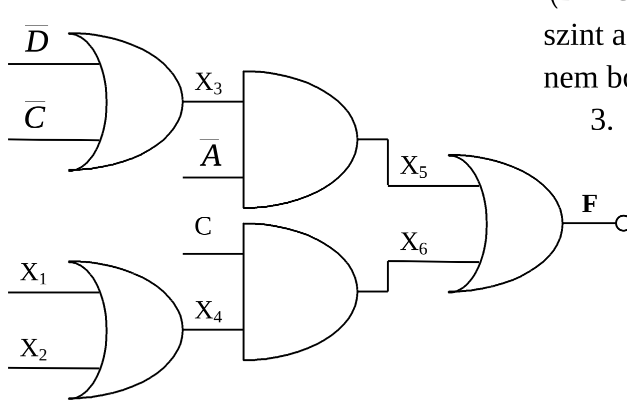
Nézzük meg most a példafüggvény szintjeit:

$$F = C \cdot \left(D \cdot \overline{B} \cdot A + \overline{\overline{D} \cdot B}\right) + \left(\overline{D} + \overline{C}\right) \cdot \overline{A}$$




Felülről lefelé realizálás:

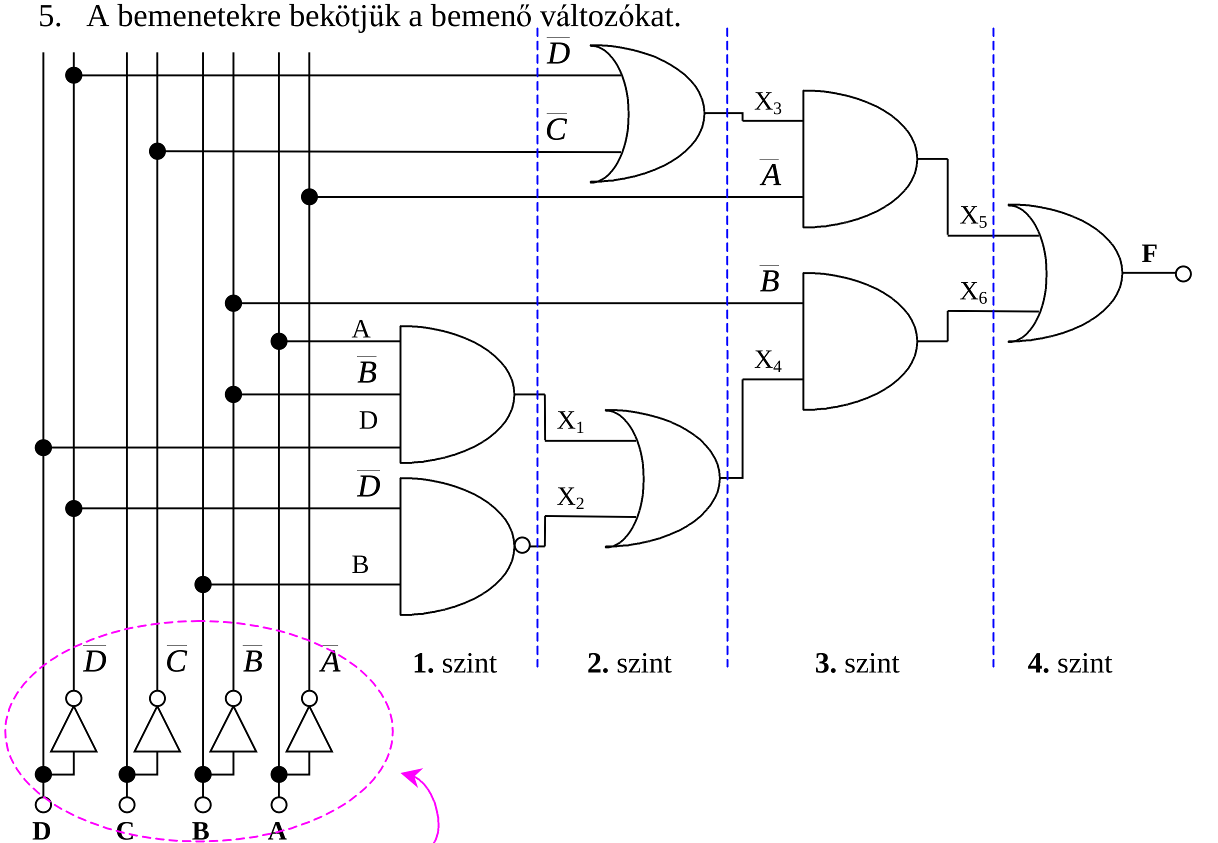
Mindig azt a logikai műveletet valósítjuk meg sorrendben előbb, ami a legmagasabb szinten van.

1. F fügvényben az OR művelet az első ilyen művelet, a logikai egységek $\left(\overline{D} + \overline{C}\right) \cdot \overline{A} = X_5$ és $C \cdot \left(D \cdot \overline{B} \cdot A + \overline{\overline{D} \cdot B}\right) = X_6$. Így: $F = C \cdot \left(D \cdot \overline{B} \cdot A + \overline{D} \cdot B\right) + \left(\overline{D} + \overline{C}\right) \cdot \overline{A} = X_5 + X_6$
2. Bontsuk fel $X_5$-t és $X_6$-t. $X_5$-ben a következő legmagasabb szint az AND művelet. A logikai egységek $\overline{A}$, ez nem bontható tovább és $\left(\overline{D} + \overline{C}\right) = X_3$. $X_6$-ben a következő legmagasabb szint az AND művelet. A logikai egységek C, ez nem bontható tovább és $\left(D \cdot \overline{B} \cdot A + \overline{\overline{D} \cdot B}\right) = X_4$.
   3. $X_3$-ban a következő legmagasabb szint az OR művelet. A logikai egységek $\overline{D}$ és $\overline{C}$, ezek nem bonthatók tovább. $X_4$-ben a következő legmagasabb szint az OR művelet. A logikai egységek $D\overline{B}A = X_1$ és $\overline{\overline{D} \cdot B} = X_2$.



PDF created with FinePrint pdfFactory trial version http://www.fineprint.com

4. $X_1$-ben a következő legmagasabb szint az AND művelet. A logikai egységek $D$, $\overline{B}$ és $A$, ezek nem bonthatók tovább. $X_2$-ban a következő legmagasabb szint az NAND művelet. A logikai egységek $\overline{D}$ és $B$, ezek nem bonthatók tovább.
5. A bemenetekre bekötjük a bemenő változókat.




A fenti megoldásban a bemeneti változók negáltjait már a bemeneten előállítjuk, így egy inverter elég, akár egy akár több bemenetre kell vezetni negáltjelet.
A bemenőváltozók negálása is kaput igényel, ezért azt is önálló szintként kezeljük (ez most a 0. szint sorszámot kapja, de legközelebb a szintek megszámolásakor figyelembe kell venni és 1. szint sorszámmal kell ellátni).

Látható, hogy a felrajzolt hálózat 5 szintű. Mivel a kapuink véges sebességűek, ezért a kimenet változása a bemenetek változásához képest késik. Ez a késés a négy szinten halmozódik, ami egyrészt a hálózat lassulását, mésrészt esetenként hibás működést eredményez. Sebesség szempontjából előnyösebb a legegyszerűbb kanonikus konjunktív és diszjunktív alak, bár általában több elemből áll, de csak kettő (a bemenőváltozók negálásával együtt 3) szintből áll.

Az alapkapukat DIP tokozású (“százlábú”) integrált áramkörökkel (IC-kel) valósítják meg. Egy IC általában 4 kétbemenetű, vagy 3 3 bemenetű kaput, vagy 6 invertert tartalmaz. Ezért, ha egy függvényhez kell 1 db OR kapu és 2 db AND, és 2 db inverter, és mindegyikhez egy külön IC-t (összesen hármat) használnánk fel, akkor a maradna 3 db OR kapu, 2 db AND kapu és 4 db inverter kihasználatlan. Összetett függvényeknél ez jelentős pazarlást jelenthet, ezért sokszor egyszerűbb azonos típusú kapukból (NAND, NOR) megépíteni a függvényt megvalósító áramkört, ekvivalens átalakítások (főként De-Morgan) segítségével.

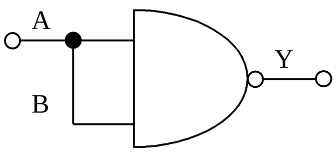


PDF created with FinePrint pdfFactory trial version http://www.fineprint.com

Csak NAND és csak NOR kapukkal bármelyik művelet megvalósítható (legfeljebb sok kell belőle).

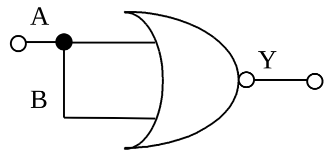
NAND-ból NOT (inverter):
Az igazságtáblából látható, ha A=B akkor $Y=\overline{A}$.

| A | B | Y | |
|---|---|---|---|
| 0 | 0 | 1 | A=B |
| 0 | 1 | 1 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | A=B |



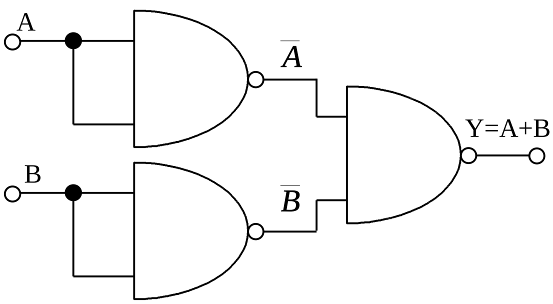

NOR-ból NOT (inverter):
Az igazságtáblából látható, ha A=B akkor $Y=\overline{A}$.

| A | B | Y | |
|---|---|---|---|
| 0 | 0 | 1 | A=B |
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 1 | 1 | 0 | A=B |



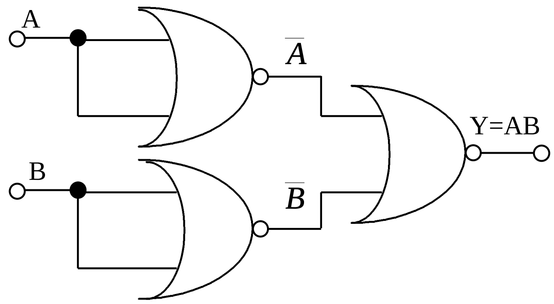

NAND-ból OR:

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




NOR-ból AND:

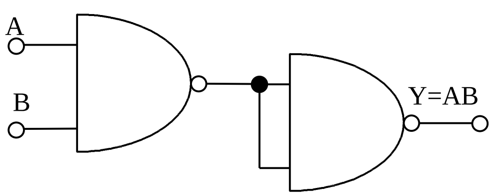
$$A\cdot B=\overline{\overline{A\cdot B}}=\overline{\overline{A}+\overline{B}}$$



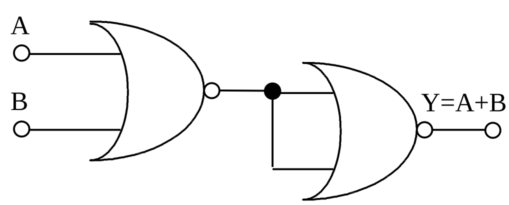

NAND-ból NOR: Mint fenn, csak a kimenet után kapcsolunk egy inverternek bekötött NAND kaput.

NOR-ból NAND: Mint fenn, csak a kimenet után kapcsolunk egy inverternek bekötött NOR kaput.

NAND-ból AND: NAND-kapu kimenete után kapcsolunk egy inverternek bekötött NAND kaput.




NOR-ból OR: NOR-kapu kimenete után kapcsolunk egy inverternek bekötött NOR kaput.




Realizáljuk pl a következő függvényt NAND kapukból:

$$F=(A+BC)\cdot\left(\overline{B+C\overline{A}}\right)$$

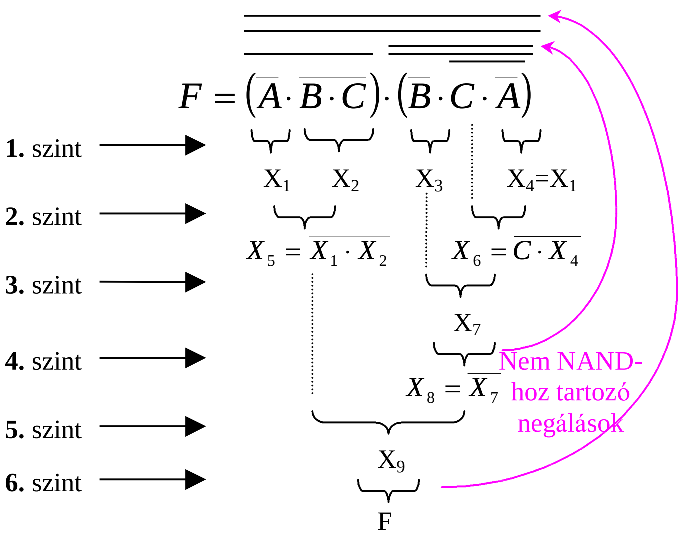
De-Morgan azonosságokkal átalakítjuk F-t, hogy csak AND és NOT művelet legyen benne:

$$F=\overline{\overline{(A+BC)}}\cdot\left(\overline{B+C\overline{A}}\right)=\left(\overline{\overline{A}\cdot\overline{B\cdot C}}\right)\cdot\left(\overline{B}\cdot\overline{C\cdot\overline{A}}\right)$$

$$F=\overline{\overline{\left(\overline{\overline{A}\cdot\overline{B\cdot C}}\right)\cdot\left(\overline{\overline{\overline{B}\cdot\overline{C\cdot\overline{A}}}}\right)}}$$

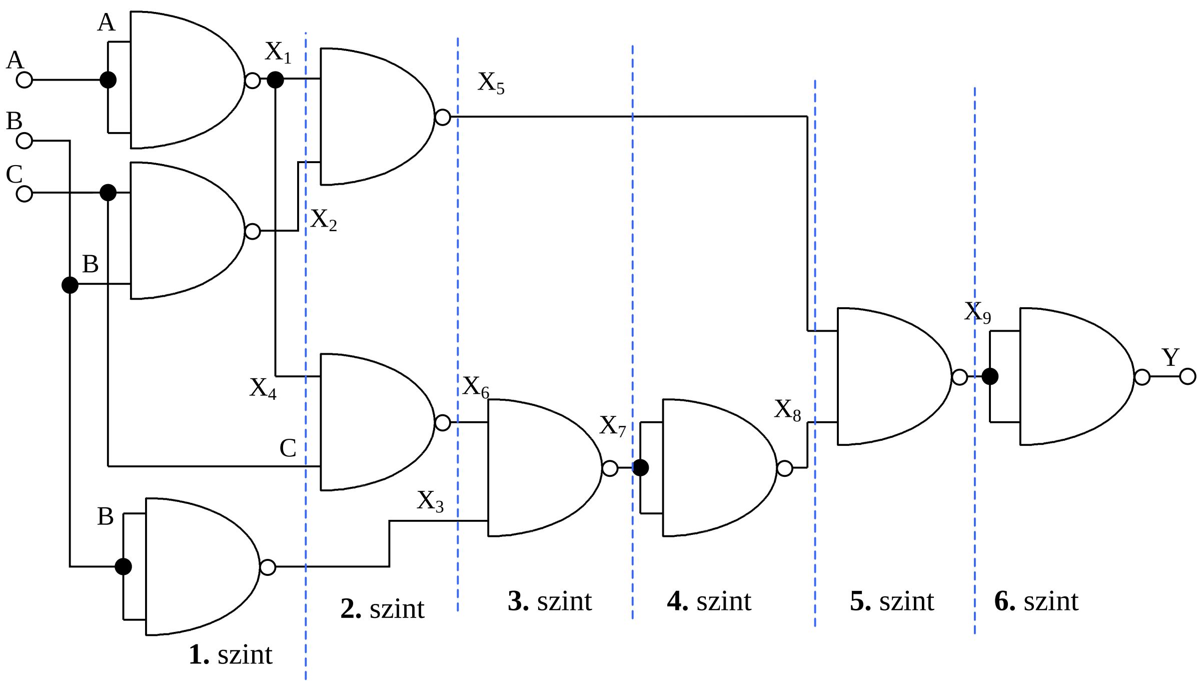


PDF created with FinePrint pdfFactory trial version http://www.fineprint.com




Mint az előbb láttuk, az AND művelet megfelel a NOT-NAND műveletnek, ezért ha AND műveletet kell elvégezni, akkor azt kétszer negáljuk, és így realizálható lesz NAND kapukkal.
A már megismert módon, először most is meghatározzuk a függvény szintjeit. Itt most egy szintet egy NAND művelet, vagy egy nem NAND művelethez tartozó negálás jelent (pl. a bemenőváltozók negálása, az AND kétszeres negálásának utolsó negálása).

Ezután elkezdhetjük felülről lefelé a realizálást. Látható, hogy $\overline{A}$ két helyen is szerepel, ezt célszerű egy inverterrel megvalósítani és onnan ágaztatni kétfelé.




Ugyanezt a függvényt realizálhatjuk NOR kapukkal is.

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

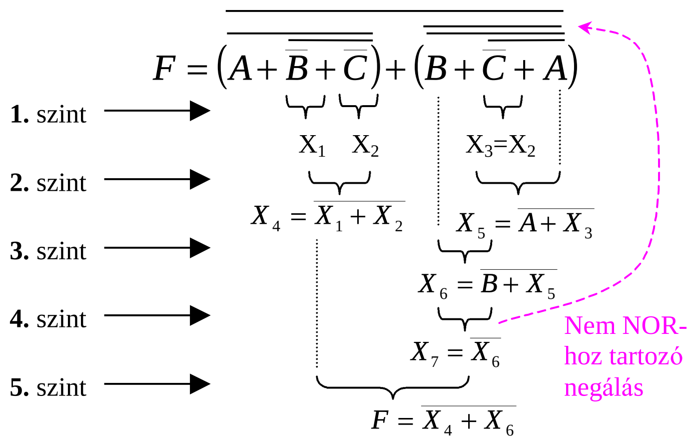
De-Morgan azonosságokkal átalakítjuk F-t, hogy csak OR és NOT művelet legyen benne:

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

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

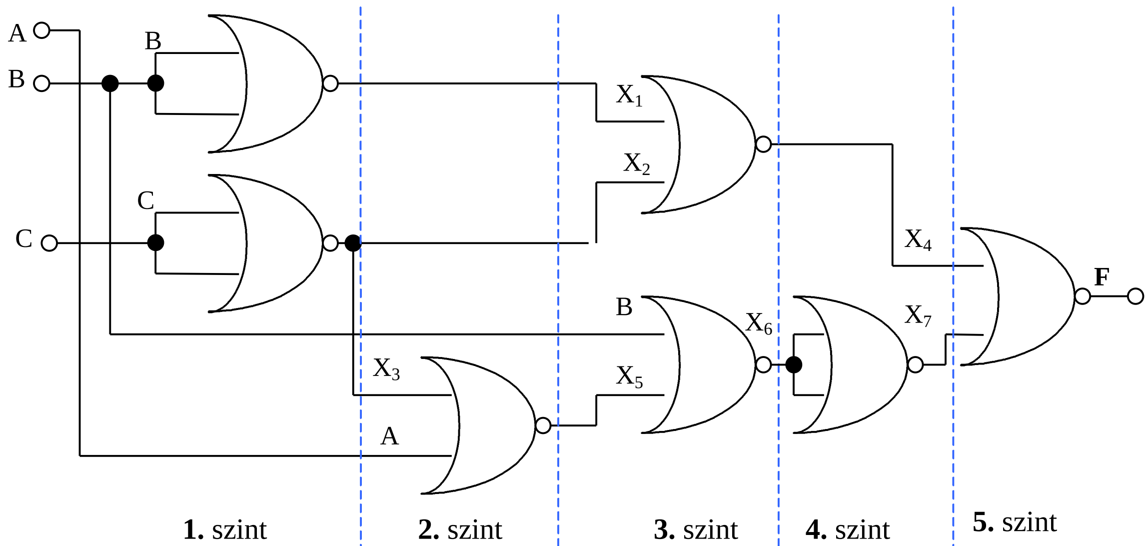


PDF created with FinePrint pdfFactory trial version http://www.fineprint.com




Mint korábban láttuk, az OR művelet megfelel a NOT-NOR műveletnek, ezért ha OR műveletet kell elvégezni, akkor azt kétszer negáljuk, és így realizálható lesz NOR kapukkal.
A már megismert módon, először most is meghatározzuk a függvény szintjeit. Itt most egy szintet egy NOR művelet vagy egy nem NOR művelethez tartozó negálás jelent (pl. a bemenőváltozók negálása, az OR kétszeres negálásának utolsó negálása).

Ezután elkezdhetjük felülről lefelé a realizálást. Látható, hogy $\overline{C}$ két helyen is szerepel, ezt célszerű egy inverterrel megvalósítani és onnan ágaztatni kétfelé.




Látható, hogy a függvény NOR kapukkal megvalósítva egy szintel kevesebb, mint NAND kapukkal.
Egy függvényt csak NOR vagy NAND kapukkal megvalósító De-Morgan átalakítását részletesebben végigvezetünk:

Legyen pl. $F = A\cdot(B + C\cdot D) + A\cdot\overline{C}$ először valósítsuk meg NOR kapukkal.
Első lépésben minden AND műveletet De-Morgan azonosságokkal OR műveletté alakítunk.

A De-Morgan azonosság szerint szerint $\overline{A\cdot B} = \overline{A} + \overline{B}$.
Ha az AND nem negált akkor kétszer megnegáljuk:

$A\cdot B = \overline{\overline{A\cdot B}} = \overline{\overline{A}+\overline{B}}$ Az első (fekete vonal) negálás azért kell, hogy De-Morgannal át tudjuk alakítani; a második (kék vonal) azért, hogy ne a negált-műveletet állítsuk elő.

Tehát az első lépések:

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

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



PDF created with FinePrint pdfFactory trial version http://www.fineprint.com

Az erdményül kapott függvényben már csak OR műveletek vannak és negálások. A következő lépésben az OR műveletekből NOR-t csinálunk két egymás utáni negálással. (Mert: A OR B = NOT(A NOR B) )
A negálást NOR kapuval meg tudjuk valósítani, ha a kapu összes bemenetére bevezetjük a negálandó jelet ( $\overline{A+A} = \overline{A}\cdot\overline{A} = \overline{A}$ ).
Az eredményül kapott függvényben most csak egy OR műveletet kell kétszeres negálással átalakítani:

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

Az átalakított OR művelet

Az első (zöld vonal) negálás azért kell, hogy NOR-á át tudjuk alakítani; a második (piros vonal) azért, hogy ne a negált-műveletet állítsuk elő.

Most valósítsuk meg NAND kapukkal.
Első lépésben minden OR műveletet De-Morgan azonosságokkal AND műveletté alakítunk.
A De-Morgan azonosság szerint szerint $\overline{A+B} = \overline{A}\cdot\overline{B}$.
Ha az OR nem negált akkor kétszer megnegáljuk:

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

Az első (fekete vonal) negálás azért kell, hogy De-Morgannal át tudjuk alakítani; a második (kék vonal) azért, hogy ne a negált-műveletet állítsuk elő.

Tehát az első lépések:

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

$$F = \overline{\overline{A\,\overline{\overline{(\overline{B}\;\;\overline{C\,D})}} + A\,\overline{C}}} \longrightarrow F = \overline{\overline{A\,\overline{\overline{(\overline{B}\;\;\overline{C\,D})}}}\quad\overline{A\,\overline{C}}}$$

Az erdményül kapott függvényben már csak AND műveletek vannak és negálások. A következő lépésben az AND műveletekből NAND-t csinálunk két egymás utáni negálással. (Mert: A AND B = NOT(A NAND B) )
A negálást NOR kapuval meg tudjuk valósítani, ha a kapu összes bemenetére bevezetjük a negálandó jelet ( $\overline{A\cdot A} = \overline{A}+\overline{A} = \overline{A}$ ).
Az eredményül kapott függvényben most minden művelet NAND, így azt nem kell tovább alakítani.



PDF created with FinePrint pdfFactory trial version http://www.fineprint.com