

Logikai szorzás, ÉS kapcsolat azonosságai:

- $0 \cdot 0 = 0$
- $0 \cdot 1 = 0$
- $1 \cdot 0 = 0$
- $1 \cdot 1 = 1$
- $A \cdot 1 = A$
- $A \cdot 0 = 0$
- $AA = A$ (idempotencia)
- $A\bar{A} = 0$
- $AB = BA$ (kommutativitás)
- $(AB)C = A(BC) = ABC$ (asszociativitás)

Logikai összeadás, VAGY kapcsolat azonosságai:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 1$
- $A + 0 = A$
- $A + 1 = 1$
- $A + A = A$ (idempotencia)
- $A + \bar{A} = 1$
- $A + B = B + A$ (kommutativitás)
- $(A + B) + C = A + (B + C) = A + B + C$ (asszociativitás)

Logikai tagadás, NEM (negáció) azonosságai:

- $\bar{0} = 1$
- $\bar{1} = 0$
- $\bar{\bar{A}} = A$

További azonosságok:

- $A(B + C) = AB + AC$ (disztributivitás)
- $A + BC = (A + B)(A + C)$ (disztributivitás)
- $A + AB = A$ (elnyelési tulajdonság)
- $A(A + B) = A$ (elnyelési tulajdonság)
- $AB + A\bar{B} = A$
- $(A + B)(A + \bar{B}) = A$
- $A + \bar{A}B = A + B$
- $A(\bar{A} + B) = AB$

De-Morgan azonosságok

- $\overline{A + B} = \bar{A}\bar{B}$
- $\overline{A + B + C} = \bar{A}\bar{B}\bar{C}$
- ...
- $\overline{AB} = \bar{A} + \bar{B}$
- $\overline{ABC} = \bar{A} + \bar{B} + \bar{C}$
- ...

Algebrai módszer

1. Példa:

Egy logikai kifejezést több féle képen is le lehet egyszerűsíteni. Legyen $\overline{AB} + \overline{AD} + \overline{AB}$ az egyszerűsítendő kifejezés. Első lépésben a De-Morgan azonosság segítségével eltüntethető a teljes kifejezés negálása. Eredményül három negált minterm ÉS kapcsolata adódik: $\overline{(AB)} \overline{(AD)} \overline{(AB)}$. Ahhoz, hogy a mintermek ne legyenek negálva, ezeken is külön-külön alkalmazni kell a De-Morgan azonosságot. Ez a függvény konjunktív normál formáját eredményezi: $(\overline{A} + \overline{B})(\overline{A} + \overline{D})(\overline{A} + \overline{B})$. Következő lépésben a disztributivitás tulajdonságát kihasználva össze lehet vonni a második két zárójelet: $(\overline{A} + \overline{B})(\overline{A}\overline{A} + \overline{A}\overline{D} + \overline{A}\overline{B} + \overline{B}\overline{B})$. Ezután kihasználható, hogy egy logikai változó ÉS kapcsolata a változó negáltjával 0-át és egy logikai változó VAGY kapcsolata 1-el a változót adja eredményül: $(\overline{A} + \overline{B})(\overline{A}\overline{B} + \overline{A}\overline{D})$. Ötödik lépésben a disztributivitást ismét felhasználva felbonthatóak a zárójelek: $\overline{A}\overline{A}\overline{B} + \overline{A}\overline{A}\overline{D} + \overline{A}\overline{B}\overline{B} + \overline{A}\overline{B}\overline{D}$. Ezután ismét felhasználható az előző tulajdonság valamint az, hogy bármely kifejezés ÉS kapcsolata 0-val 0: $\overline{A}\overline{A}\overline{B} + \overline{A}\overline{B}\overline{D}$. Az ÉS művelet idempotencia tulajdonsága alapján mind a két szorzatot egyszerűsíthető: $\overline{A}\overline{B} + \overline{A}\overline{B}$. Végül a VAGY művelet idempotencia tulajdonsága alapján ismét egyszerűsíteni lehet: $\overline{A}\overline{B}$. Az egyszerűsítés teljes folyamata alább látható.

$$\begin{aligned} \overline{AB} + \overline{AD} + \overline{AB} &= \\ \overline{(AB)} \overline{(AD)} \overline{(AB)} &= \\ (\overline{A} + \overline{B})(\overline{A} + \overline{D})(\overline{A} + \overline{B}) &= \\ (\overline{A} + \overline{B})(\overline{A}\overline{A} + \overline{A}\overline{D} + \overline{A}\overline{B} + \overline{B}\overline{B}) &= \\ (\overline{A} + \overline{B})(\overline{A}\overline{B} + \overline{A}\overline{D}) &= \\ \overline{A}\overline{A}\overline{B} + \overline{A}\overline{A}\overline{D} + \overline{A}\overline{B}\overline{B} + \overline{A}\overline{B}\overline{D} &= \\ \overline{A}\overline{A}\overline{B} + \overline{A}\overline{B}\overline{D} &= \\ \overline{A}\overline{B} + \overline{A}\overline{B} &= \\ \overline{A}\overline{B} & \end{aligned}$$

Ugyanez a logikai függvény másként is egyszerűsíthető, most más sorrendben használva a Boole algebrai azonosságokat. Első lépésben kihasználható a disztributivitás tulajdonság így az első két mintermből kiemelhető $\overline{B}$: $\overline{(A + \overline{A})B} + \overline{A}\overline{B}$. A logikai VAGY kapcsolat idempotencia tulajdonsága alapján az $\overline{(A + \overline{A})}$ kifejezés kiesik: $\overline{B} + \overline{A}\overline{B}$. Következő lépésben az $A + \overline{A}B = A + \overline{B}$ azonosságot felhasználva a kifejezés átalakítható: $\overline{B} + \overline{A}$. Majd utolsó lépésben egy De-Morgan azonosság segítségével megkapható a végső formát: $\overline{A}\overline{B}$. Az egyszerűsítés teljes menete alább látható.

$$\begin{aligned} \overline{AB} + \overline{AD} + \overline{AB} &= \\ \overline{(A + \overline{A})B} + \overline{A}\overline{B} &= \\ \overline{B} + \overline{A}\overline{B} &= \\ \overline{B} + \overline{A} &= \\ \overline{A}\overline{B} & \end{aligned}$$

Kifejtési módszer

Az algebrai módszer komplexebb, több változót tartalmazó logikai kifejezések $F(x_1, x_2, \dots, x_n)$ egyszerűsítése esetén nehezen alkalmazható. Ilyen esetekben alkalmazható a kifejtési módszer. Ekkor egy adott változó értékét először ponáltnak $F(1, x_2, \dots, x_n)$, majd

negálnak $F(0, x_2, \dots, x_n)$ kell definiálni. Ezután a két egyszerűbb, kevesebb változót tartalmazó függvényt egyszerűsíthető (lehet rekurzívan kifejtési módszert is alkalmazni). Az eredeti függvény egyszerűsített formája kétféleképpen is megkapható.

- A változó ponált rögzítésével kapott függvényhez ÉS logikai kapcsolattal hozzá kell fűzni a változót ponált értékben, a negált rögzítéssel kapott függvényhez pedig negált értékben. Végül az így kiszámított két logikai kifejezést VAGY logikai művelettel kell összekötni.

$$F(x_1, x_2, \dots, x_n) = x_1 \cdot F(1, x_2, \dots, x_n) + \overline{x_1} \cdot F(0, x_2, \dots, x_n)$$

- A változó negált rögzítésével kapott függvény negáltjához VAGY logikai kapcsolattal kell hozzáfűzni a változót ponált értékben, a ponált rögzítéssel kapott függvény negáltjához pedig negált értékben. Végül az így kiszámított két logikai kifejezést ÉS logikai művelettel kell összekötni és negálni.

$$F(x_1, x_2, \dots, x_n) = \overline{(x_1 + F(0, x_2, \dots, x_n)) \cdot (\overline{x_1} + F(1, x_2, \dots, x_n))}$$

Legyen adott diszjunktív normál formában egy háromváltozós F függvény, ahol $F(A, B, C) = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + AB\overline{C}$. Legyen az A változó értéke először 1, ekkor $F(1, B, C) = 0\overline{B}\overline{C} + 0B\overline{C} + 1\overline{B}\overline{C} + 1B\overline{C} = \overline{B}\overline{C} + B\overline{C} = (\overline{B} + B)\overline{C} = \overline{C}$. Ezek után legyen A értéke 0, azaz $F(0, B, C) = 1\overline{B}\overline{C} + 1B\overline{C} + 0\overline{B}\overline{C} + 0B\overline{C} = \overline{B}\overline{C} + B\overline{C} = \overline{B}(\overline{C} + C) = \overline{B}$.

- Az egyszerűsített logikai függvény meghatározása az első módszer szerint, azaz $F(A, B, C) = A \cdot F(1, B, C) + \overline{A} \cdot F(0, B, C) = A\overline{C} + \overline{A}\overline{B}$.
- Az egyszerűsített logikai függvény meghatározása a második módszer szerint, azaz $F(A, B, C) = \overline{(A + F(1, B, C)) \cdot (\overline{A} + F(0, B, C))} = \overline{(A + \overline{C}) \cdot (\overline{A} + \overline{B})} = \overline{(\overline{A} + \overline{B})(A + \overline{C})} = \overline{A}\overline{B} + A\overline{C}$.

Kifejtési módszer esetében fontos, hogy melyik változó kerül rögzítésre, ennek hatékony kiválasztása az algebrai módszerhez hasonlóan nagy gyakorlatot igényel.

Grafikus módszer

Hurkok felrajzolásának szabályai

A hurkok optimális felrajzolásának, vagy más néven a tömbösítésnek, vannak szabályai, amelyeket az egyszerűsítés során be kell tartani.

- 2^n ($n = 0, 1, 2, \dots$) darab minterm (maxterm) vonható be egy hurokba.
- A hurkok téglalap alakúak.
- Minden 1-est (maxtermek esetében 0-ást) tartalmaznia kell legalább egy huroknak és 0 (maxterm esetben 1) nem kerülhet hurokba.
- Egyik hurok a másikat nem tartalmazhatja teljes mértékben.
- Egy term több hurokban is szerepelhet.
- A hurkok felrajzolásánál törekedni kell a lehető legnagyobb méretű hurkok használatára.
- Közömbös ('-') kimeneti függvényértékeket a jobb (optimálisabb) lefedésnek megfelelően kell megválasztani (NTSH).

2. Példa

Készíteni kell egy logikai hálózatot, amely bemeneteihez egy súly tartozik és az F kimeneten akkor lesz 1 logikai érték, ha a bemenetekösszűlya nagyobb, mint 1 azon bemenetekre, ahol a logikai érték 0. Az A bemenet súlya legyen 3, a B bemeneté 2, a C és a D bemeneteké pedig 1. Az igazságtáblázatot ez alapján fel lehet írni.

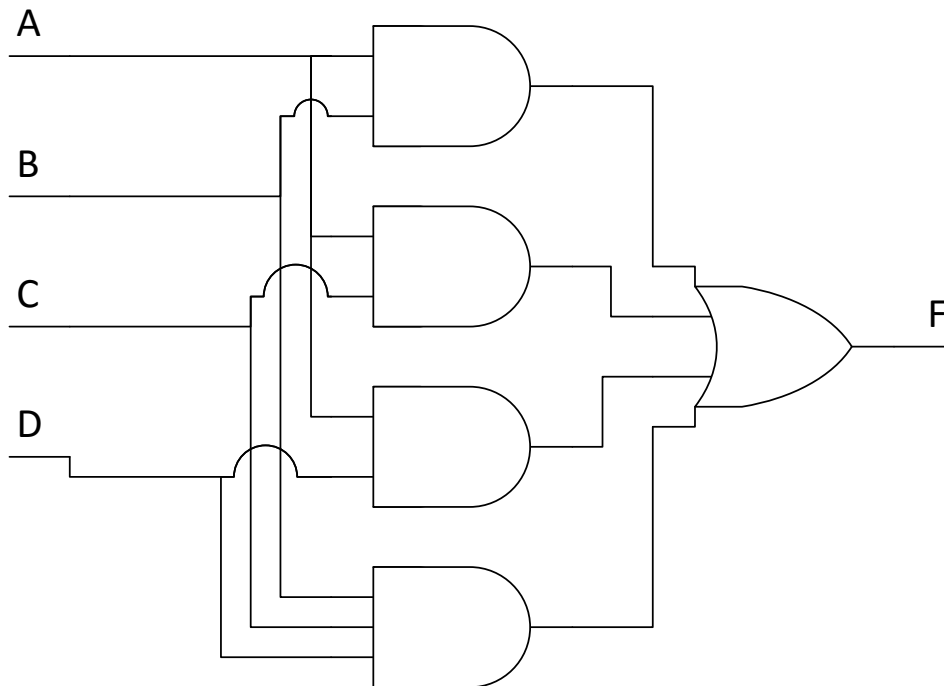
A	B	C	D	F
0	0	0	0	0

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

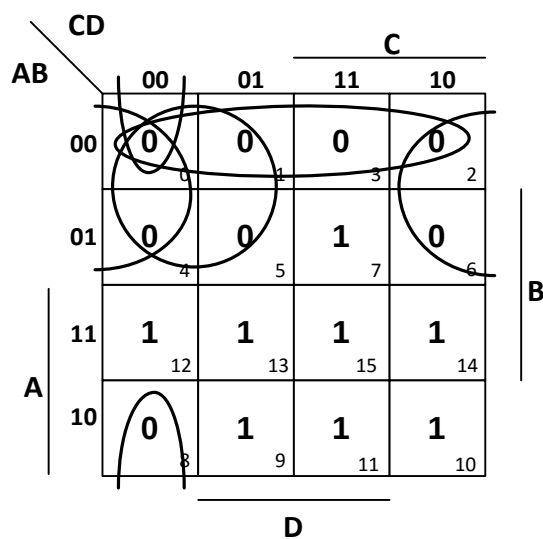
Az igazságtáblázatból fel lehet rajzolni a Karnaugh táblát. Először a diszjunktív alak meghatározása a cél. A prímiplikánsok meghatározhatók a Karnaugh tábla segítségével. Ebben az esetben az összes megtalált prímiplikáns lényeges prímiplikáns.

		CD					
				C			
AB		00	01	11	10		
A	00	0 0	0 1	0 3	0 2		
	01	0 4	0 5	1 7	0 6		
	11	1 12	1 13	1 15	1 14		
	10	0 8	1 9	1 11	1 10		
		D					

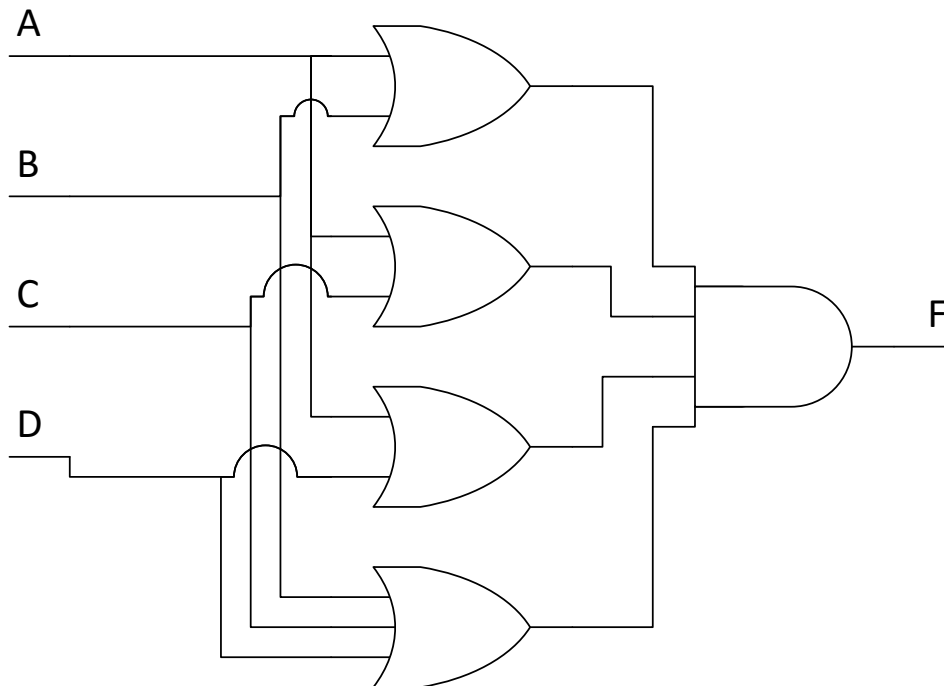
Mivel az összes lényeges prímiplikánst tartalmaznia kell az egyszerűsített függvényalaknak, ezért az egyszerűsített függvényalak a következő $F = AB + AC + AD + BCD$, melynek a megvalósítása az alábbi ábrán látható.



Konjunktív alak meghatározása esetén a Karnaugh táblában a 0-kat kell hurkolni a prímmimplikánsok megtalálásához.



Az itt megtalált összes prímmimplikáns is lényeges prímmimplikáns, ezért nem hagyható el a megoldásból. Az eredményül kapott $F = (A + B)(A + C)(A + D)(B + C + D)$ logikai áramköri realizációja alább látható.



A megoldásként kapott diszjunktív és konjunktív alak ugyanynyi kapubemenetet tartalmaz, ezért mind a kettő ugyanolyan egyszerűnek számít. Ezek után a rendelkezésre álló konkrét logikai áramkörök tulajdonságai (ár, rendelkezésre állás, méret, ...) határozzák meg, hogy melyiket kell megvalósítani.

Nem teljesen specifikált hálózatok minimalizálása

Egy példán keresztül bemutatásra kerül, hogy a határozatlan állapotok 0-nak vagy 1-nek választása mennyiben befolyásolja a megvalósítandó logikai kapcsolási vázlatot. A következő igazságtáblában nyolc darab 1-es és kettő határozatlan állapot szerepel. Karnaugh tábla segítségével meghatározhatók a prímisszimplikánsok.

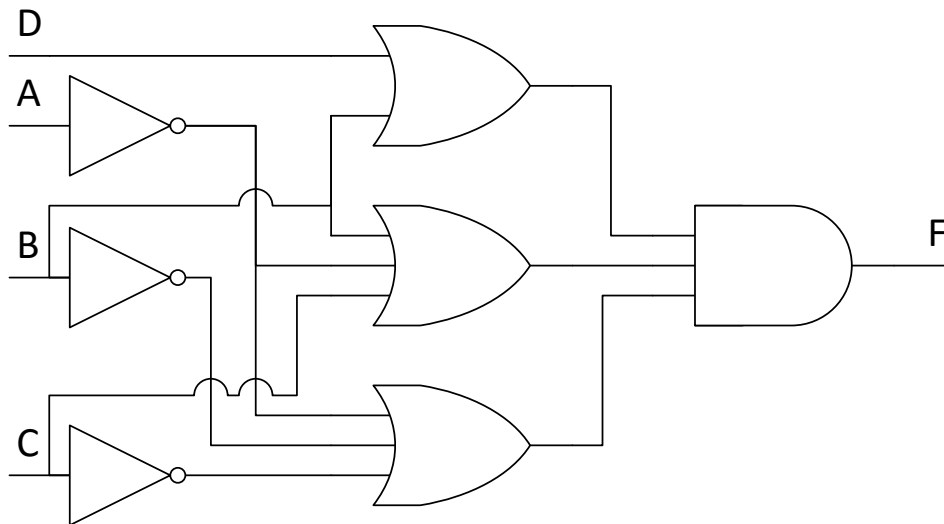
A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	-
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	-
1	1	1	1	1

Először megvizsgálásra kerül, hogy mi történik, ha minden határozatlan állapotot 0. Ekkor az igazságtábla a következő képen módosul.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

A Karnaugh tábla három darab kettes és egy négyes hurkot eredményez. A Karnaugh táblából a korábbiaknak megfelelően meghatározható az egyszerűsített függvényalak, mely három prímimplikánst tartalmaz és a következő függvényt eredményezi $F = BD + \overline{A}BC + A\overline{B}C$.

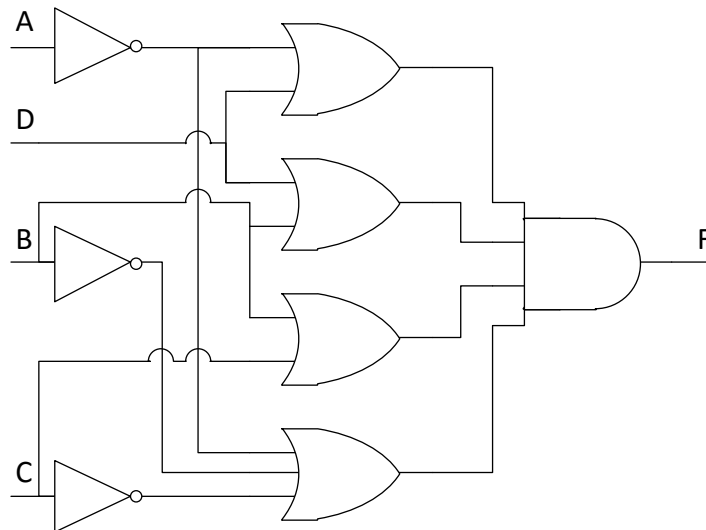
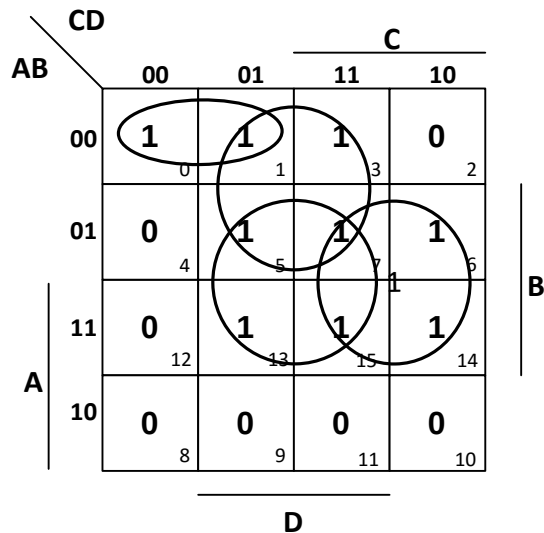
		CD			
		00	01	11	10
AB	00	1 ₀	1 ₁	0 ₃	0 ₂
	01	0 ₄	1 ₅	1 ₇	1 ₆
	11	0 ₁₂	1 ₁₃	1 ₁₅	0 ₁₄
	10	0 ₈	0 ₉	0 ₁₁	0 ₁₀
		D			
		B			



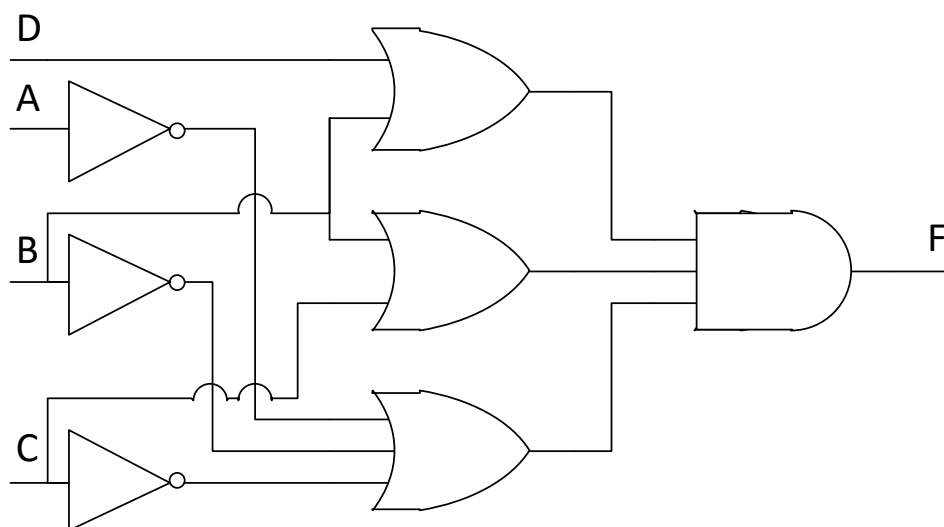
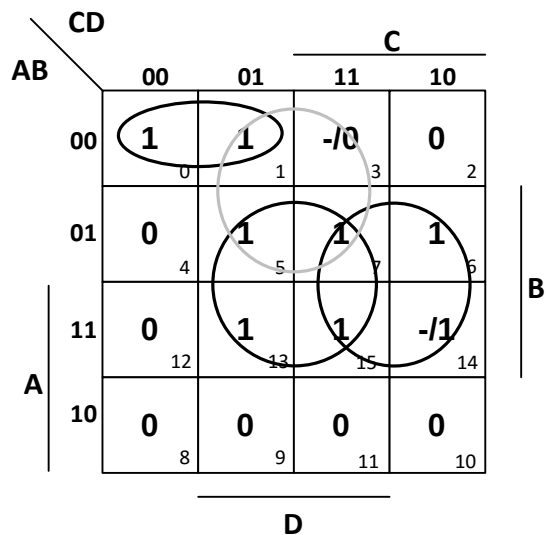
Következő esetben minden határozatlan állapot 1.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Ebben az esetben egyszerűbb prímiplikánsokat sikerül elérni, így ez egy darab kettes és három darab négyes hurkot eredményez. Az egyszerűsített függvényalak a következő $F = \bar{A}D + DD + DC + \bar{A}BC$. A prímiplikánsok valóban egyszerűbbek, kevesebb mintermet tartalmaznak, de több prímiplikánsra van szükség a függvény megvalósításához.



Egyik előző eset sem eredményezett optimális megoldást. A fejezet elején leírt módszert alkalmazva ugyanazok a hurkok keletkeznek, abban az esetben, amikor a határozatlan állapotok 1-esek, de most a prímmimplikánsok kiválasztásánál a határozatlan állapotokat ne kelljen feltétlenül lefedni. Így a következő ábrán szürkével jelölt (1, 3, 5, 7) hurkot nem kell az egyszerűsített logikai függvénynek tartalmaznia, mivel annak az 1-es, 5-ös és 7-es cellája le van fedve másik hurokkal is, a negyedik cellája viszont határozatlan állapot. Az egyszerűsített függvényalak az $F = BD + DC + \overline{A}B\overline{C}$ függvény, amely kevesebb prímmimplikánst tartalmaz, mint ha minden határozatlan állapot 1-es és egyszerűbb, mint ha 0.



Ez a megvalósítás a következő igazságtáblázatnak felel meg, amely nem mond ellent az eredeti specifikációval.

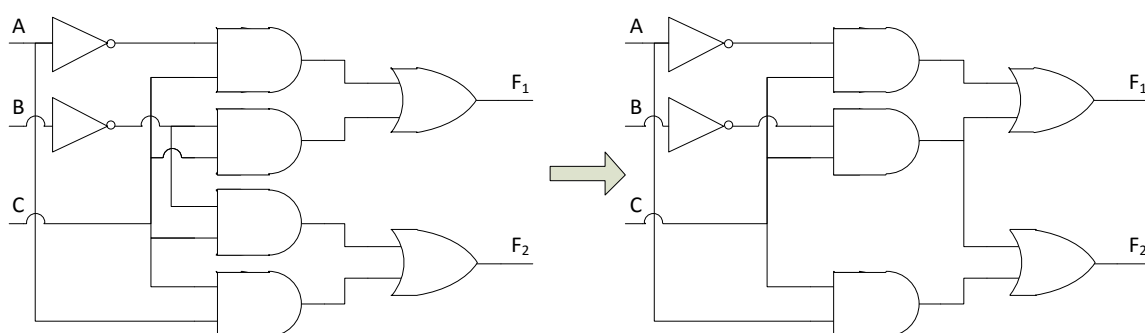
A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0

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

1. Többkimenetű logikai függvények egyszerűsítése

Többkimenetű kombinációs hálózatok esetén minden kimenethez külön logikai függvény tartozik. A logikai függvények egyszerűsítését végre lehet hajtani külön-külön egyszerűsítve őket így jutva el a legegyszerűbb diszjunktív vagy konjunktív alakhoz. Mivel mindegyik függvényhez ugyanazok a bemeneti változók tartoznak, ezért további egyszerűsítésre is lehetőség nyílik, a közös részeket nem kell többször megvalósítani.

Legyen egy három bemenetű két kimenetű logikai hálózat, ahol a kimeneti függvények legegyszerűbb diszjunktív alakja a következő $F_1 = \bar{A}C + \bar{B}C$ és $F_2 = AC + \bar{B}C$. A két függvényt külön-külön is meg lehet valósítani, de észrevehető, hogy mind a kettő tartalmazza a $\bar{B}C$ prímisszorzót, azaz ezt a logikai kapcsolatot elegendő egyszer megvalósítani. Mind a két megvalósítás logikai áramköri realizációja az alábbi ábrán látható. A második megvalósítás természetesen egyszerűbb, azt kell megvalósítani.



A cél a továbbiakban a közös prímisszorzók megtalálása, hogy azokat felhasználva kevesebb áramköri elemmel legyen megvalósítható a hálózat.

Grafikus minimalizálás

A minimalizálási eljárást módosítani kell úgy, hogy a közös prímisszorzókat csak egyszer kelljen megvalósítani. Az egyik legegyszerűbb megoldás a szorzatfüggvény vizsgálata. A szorzatfüggvény ott vesz fel 1 értéket, ahol mind a két függvény 1 értéket vesz fel, azaz a szorzatfüggvény Karnaugh táblájában ott lesz 1, ahol mind a két Karnaugh táblában 1 volt.

Az előző példát nézve a szorzatfüggvény Karnaugh táblájában egyetlen hurok szerepel, mégpedig a közös $\bar{B}C$ hurok.

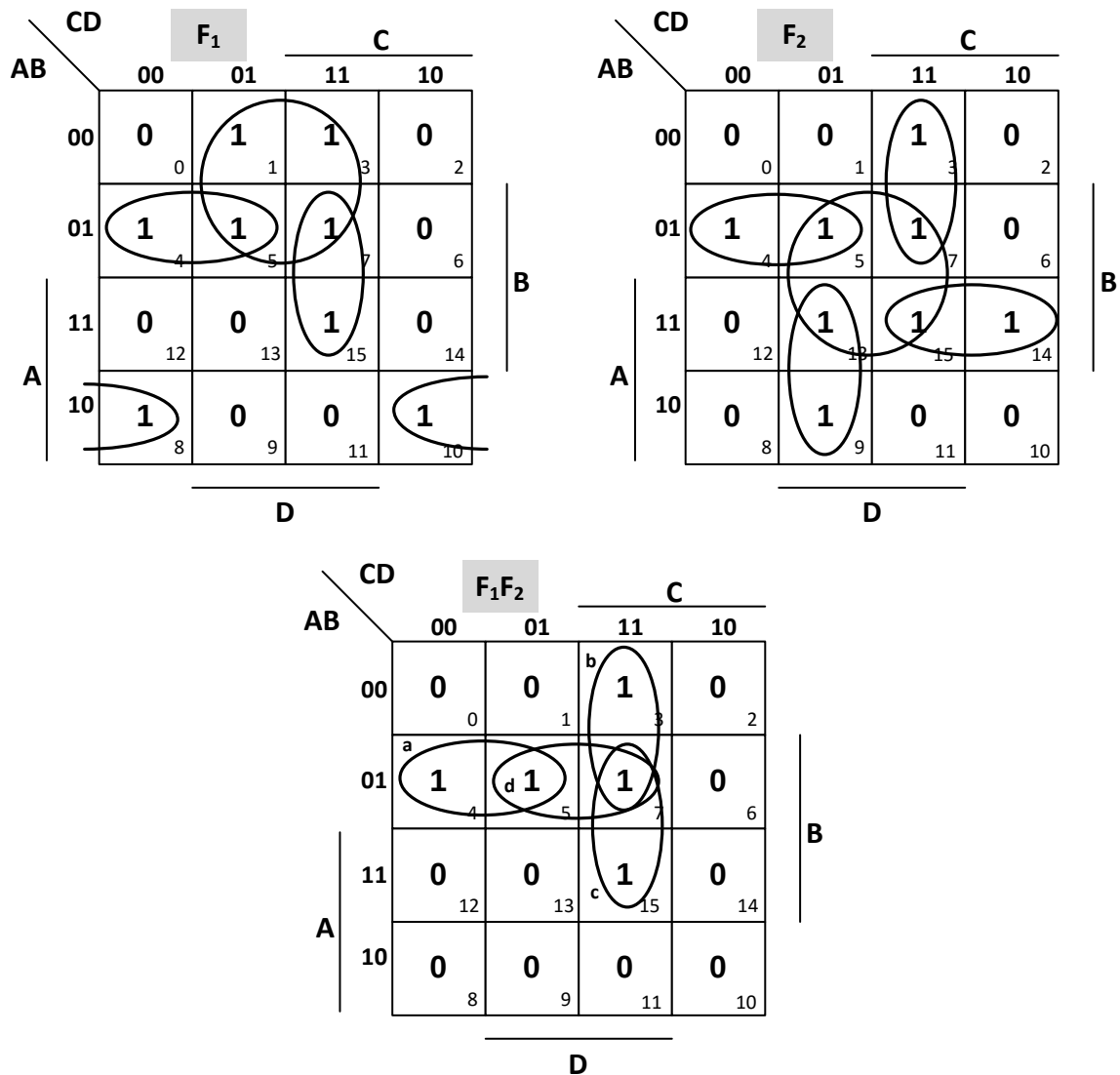
BC		C				B	F ₁
		00	01	11	10		
A	0	0	1	1	0		
	1	0	1	0	0		

BC		C				B	F ₂
		00	01	11	10		
A	0	0	1	0	0		
	1	0	1	1	0		

BC		C				B	F ₁ F ₂
		00	01	11	10		
A	0	0	1	0	0		
	1	0	1	0	0		

A szorzatfüggvénynek természetesen lehet olyan prímiplikánsa is, melyek nem prímiplikánsa valamely vagy mindegyik eredeti függvényeknek. Az ilyen prímiplikánsokat nem lehet figyelmen kívül hagyni a szorzatfüggvény optimális lefedéséhez.

A következő példa legyen egy négy bemenetű két kimenetű logikai hálózat, melynek Karnaugh táblái az alábbiakban adottak. Továbbá adott a szorzatfüggvény Karnaugh táblája is, ahol az implikánsok betűkkel vannak megjelölve a későbbi könnyebb hivatkozás érdekében.



Megvizsgálva a Karnaugh táblákat megállapítható, hogy a szorzatfüggvény prímiimplikánsai között vannak közös és nem közös prímiimplikánsok. Az **a** prímiimplikáns közös, mivel F_1 -ben és F_2 -ben is szerepel. A **b** prímiimplikáns nem közös, mivel F_2 -ben ugyan prímiimplikáns, de F_1 -ben nem prímiimplikáns, hanem egy másik prímiimplikánsnak csak egy része, azaz tovább egyszerűsíthető implikánsként szerepel. A **c** prímiimplikáns szintén nem közös, mivel F_1 -ben ugyan prímiimplikáns, de F_2 -ben nem prímiimplikáns, hanem egy másik prímiimplikánsnak csak egy része, azaz tovább egyszerűsíthető implikánsként szerepel. A **d** prímiimplikáns nem közös, mivel sem F_1 -ben sem F_2 -ben nem prímiimplikáns, hanem egy másik prímiimplikánsnak csak egy része, azaz tovább egyszerűsíthető implikánsként szerepel.

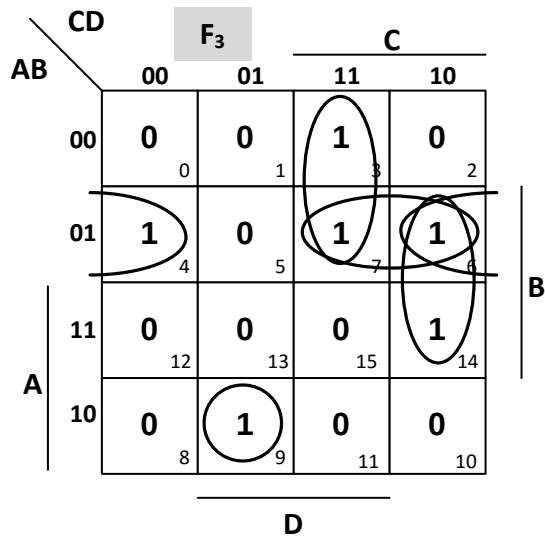
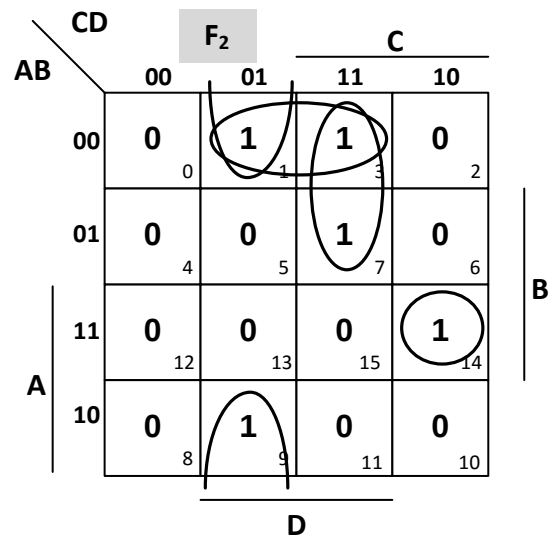
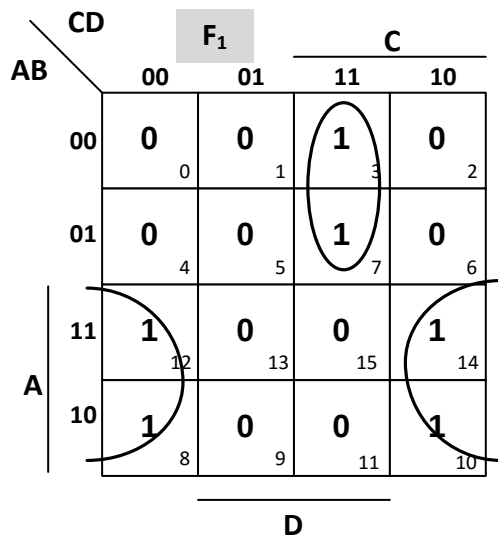
Az egyszerűsítés során a hálózat kimeneti függvényeinek összes lehetséges kombinációjára képezni kell a szorzatfüggvényt, majd ezek után minden függvényhez és szorzatfüggvényhez meg kell határozni a prímiimplikánsokat. A minimalizált függvények felírásához az összes generált prímiimplikánst figyelembe kell venni a függvények és a szorzatfüggvények prímiimplikánsait is.

Mivel a Karnaugh táblákat nehéz lehet áttekinteni, ezért a prímiimplikáns tábla valamint segédfüggvény segítségével kell kiválasztani a szükséges prímiimplikánsokat. A prímiimplikáns tábla tartalmazza az összes megtalált prímiimplikánst a függvények és függvénysszorzatok szerint csoportosítva. A táblázat kitöltését a szorzatfüggvények prímiimplikánsaival kell

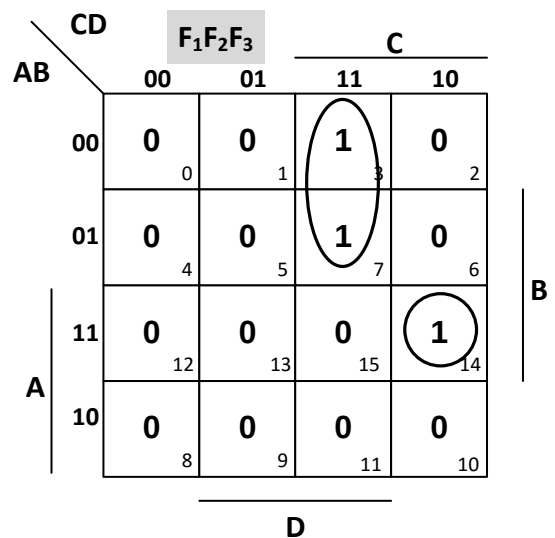
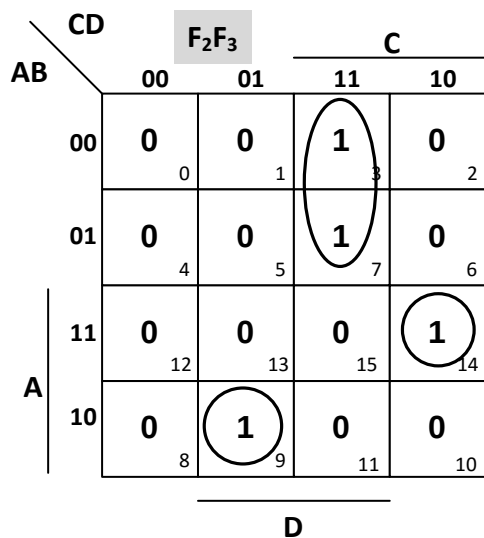
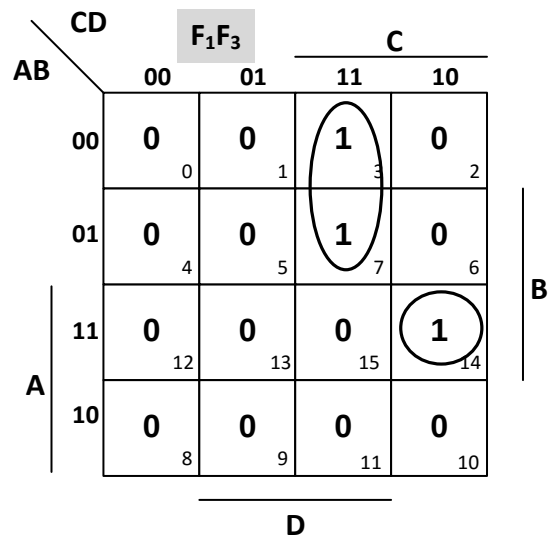
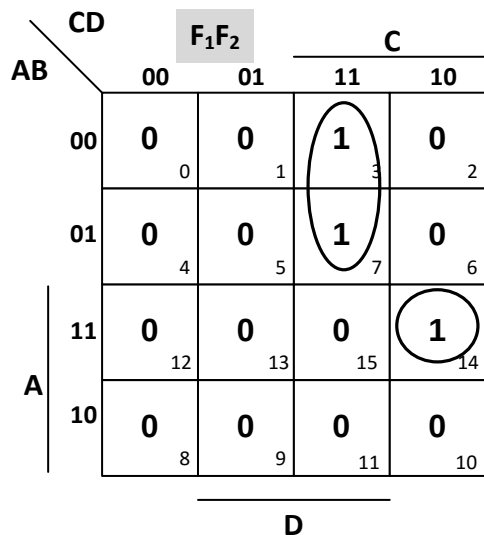
kezdeni. Ha egy szorzatfüggvénynek vagy egy függvénynek már van olyan prímisszorzat, amely szerepel a táblázatban, akkor azt nem kell még egyszer felvenni. Ha már minden prímisszorzat szerepel, akkor az a függvény vagy szorzatfüggvény nem jelenik meg külön a táblázatban. Így minden prímisszorzat csak egyszer fog szerepelni a táblázatban.

Amennyiben a szorzatfüggvény alapján több olyan megoldás létezik, melyekben ugyanannyi, minimális számú prímisszorzat szerepel, akkor ezek közül próbálgatással lehet kiválasztani az optimálist. A próbálgatás során csak a legkevesebb tényezőt tartalmazó szorzatokat kell figyelembe venni. A kiválasztott szorzatban szereplő prímisszorzatok az elvi logikai rajzban ÉS kapukkal lesznek megvalósítva, majd ezeknek az ÉS kapuknak a kimeneteit egy VAGY kapuk bemeneteire kell vezetni. Mivel azonos számú prímisszorzat szerepel mindegyik szorzatban, ezért a VAGY szinten ugyanannyi kapubemenet van, tehát az ÉS szinten kell minimalizálni a kapubemenetek számát.

A következő példában a feladat egy négy bemenetű és három kimenetű kombinációs hálózat minimális realizációjának meghatározása diszjunktív formában. Legyen a három kimeneti függvény diszjunktív normál formában adott a következőképpen. $F_1 = \sum_{i=0}^{15} (3,7,8,10,12,14)$, $F_2 = \sum_{i=0}^{15} (1,3,7,9,14)$ és $F_3 = \sum_{i=0}^{15} (3,4,6,7,9,14)$. Első lépésben meg kell határozni a kimeneti függvények prímisszorzatokat Karnaugh táblák segítségével, mely alapján $F_1 = A\bar{D} + \bar{A}CD$, $F_2 = \bar{A}\bar{B}D + \bar{A}CD + B\bar{C}D + ABC\bar{D}$ és $F_3 = \bar{A}CD + \bar{A}DC + AB\bar{D} + BCD + AB\bar{C}D$.



Második lépésben az összes lehetséges szorzatfüggvény prímisszimplikánsainak meghatározása következik, mely alapján $F_1 F_2 = \bar{A}CD + ABC\bar{D}$, $F_1 F_3 = \bar{A}CD + ABC\bar{D}$, $F_2 F_3 = \bar{A}CD + ABC\bar{D} + A\bar{B}C\bar{D}$ és $F_1 F_2 F_3 = \bar{A}CD + ABC\bar{D}$.



A generált prímimplikánsok alapján felírható a prímimplikáns tábla. Például az $\bar{A}\bar{C}D$ prímimplikáns szerepel az $F_1F_2F_3$ szorzatfüggvényben így bekerül a táblázatban, de hiába szerepel ugyanez a prímimplikáns az összes többi függvényben és szorzatfüggvényben, nem kerül be többször a táblázatba. Valamint például az F_1F_2 szorzatfüggvény nem szerepel a táblázatban, mivel az összes prímimplikánsa már korábban bekerült.

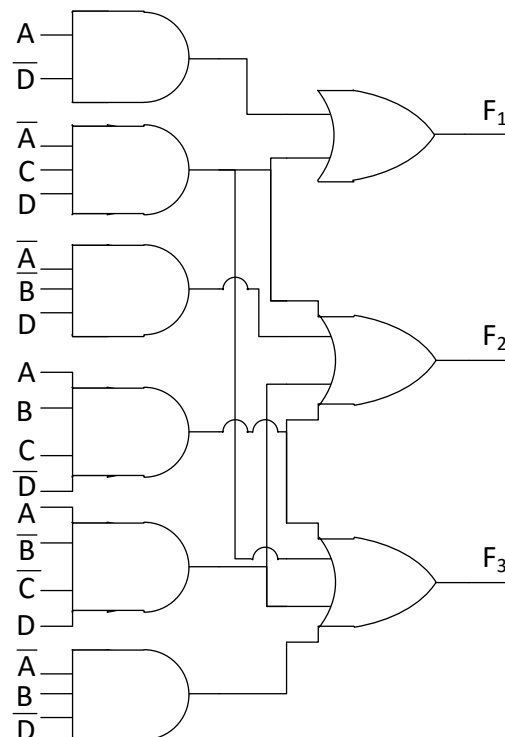
Sor	Prímimplikáns		F_1						F_2					F_3					
			3	7	8	10	12	14	1	3	7	9	14	3	4	6	7	9	14
$F_1F_2F_3$	* $\bar{A}\bar{C}D$	a	x	x						x	x			x			x		
	* $AB\bar{C}\bar{D}$	b						x				x							x
F_2F_3	* $AB\bar{C}D$	c									x							x	
F_1	* $A\bar{D}$	d			x	x	x	x											
F_2	$\bar{A}\bar{B}D$	e							x	x									
	$B\bar{C}D$	f							x			x							
F_3	$\bar{A}BC$	g													x		x		

*	$AB\bar{D}$	h				x	x	
	$BC\bar{D}$	i					x	x

A táblázatból leolvashatóak a lényeges prímiplikánsok, de ezek nem fedik le az összes mintermet, ezért mindenképpen segédfüggvényre van szükség. Mind a három függvényhez fel kell írni a segédfüggvényt, ahol $S_1 = aaddd(b + d)$, $S_2 = (e + f)(a + e)a(c + f)b$ és $S_3 = ah(g + h + i)(a + g)c(b + i)$. Az eredő segédfüggvény a segédfüggvények ÉS kapcsolata,

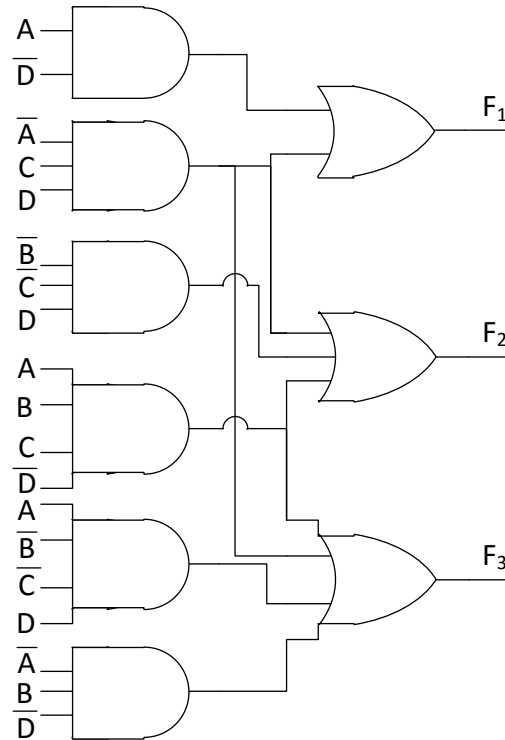
azaz $S = S_1 S_2 S_3 = aaddd(b + d)(e + f)(a + e)a(c + f)bah(g + h + i)(a + g)c(b + i)$. Az elnyelési tulajdonság segítségével a függvény egyszerűsíthető $S = abcdeh(e + f) = abcdeh + abcdfh$. A segédfüggvény két lehetséges megvalósítást eredményez, ahol mind a kettőben ugyanannyi prímiplikáns szerepel. A legegyszerűbb hálózat eléréséhez próbálgatásra van szükség, azaz mind a két lehetséges megoldást elemezni kell.

Az $abcdeh$ megvalósítása esetén mind a három függvény megvalósításához csak az adott prímiplikánsokat szabad felhasználni. Mind a három függvényhez fel kell írni egy segédfüggvény és megvizsgálni, hogy melyik függvényhez melyik prímiplikánsok kellenek, de a nem felhasználható prímiplikánsok helyett 0 értékkel kell számolni. Így $S_1 = aaddd(b + d) = ad$, $S_2 = (e + 0)(a + e)a(c + 0)b = e(a + e)ab = abce$ és $S_3 = ah(0 + h + 0)(a + 0)c(b + 0) = ahhaab = abeh$. Ebből meghatározhatók a kimenetek függvényei, $F_1 = \bar{A}CD + AD$, $F_2 = \bar{A}CD + ABC\bar{D} + A\bar{B}CD + A\bar{B}D$ és $F_3 = \bar{A}CD + ABC\bar{D} + A\bar{B}CD + A\bar{B}D$, melynek logikai áramköri realizációja lejjebb látható és a bementi negálókat nem számolva összesen 29 kapubemenetet tartalmaz.



Az $abcdfh$ megvalósítása esetén mind a három függvény megvalósításához csak az adott prímiplikánsokat szabad felhasználni. Mind a három függvényhez fel kell írni egy segédfüggvény és megvizsgálni, hogy melyik függvényhez melyik prímiplikánsok kellenek, de a nem felhasználható prímiplikánsok helyett 0 értékkel kell számolni. Így

$S_1 = aaddd(b+d) = ad$, $S_2 = (0+f)(a+0)a(c+f)b = faa(c+f)b = abf$ és
 $S_3 = ah(0+h+0)(a+0)c(b+0) = ahhaacb = abeh$. Ebből meghatározhatók a
 kimenetek függvényei, $F_1 = \bar{A}CD + AD$, $F_2 = \bar{A}CD + ABCD + \bar{B}CD$ és
 $F_3 = \bar{A}CD + ABCD + AB\bar{C}D + AB\bar{D}$, melynek logikai áramköri realizációja lejjebb látható
 és a bementi negálókat nem számolva összesen 28 kapubemenetet tartalmaz.



A két kapcsolás csak az F_2 függvény megvalósításában különbözik. A második megvalósításban egyel kevesebb prímisszám kell a függvény megvalósításához. Az első megvalósításhoz képest egy három bemenetű ÉS kapu helyett egy másik három bemenetű kell, ez az ÉS kapuk szintjén nem jelent egyszerűsítést, de a VAGY kapuk szintjén az F_2 megvalósításához négy bemenetű VAGY kapu helyett elég egy három bemenetű is, azaz egy kapubemenetet lehet megtakarítani.

Számjegyes minimalizálás

A számjegyes minimalizálás vagy más néven Quine-McCluskey módszer ugyanazt a minimalizálási alapötletet használja fel, mint a Karnaugh tábla, viszont ebben az esetben nincs korlát a változók számára. A számjegyes minimalizálás a mintermeket egyértelműen azonosító alsó indexet használja fel a számoláshoz. Ez az index egy decimális szám, amely alapján meg kell tudni mondani, hogy két minterm mikor szomszédos egymással.

Két minterm akkor szomszédos, ha csak egy változóban különböznek. Azaz ha bináris számként tekintünk rájuk, akkor csak egy helyiértékben különböznek, tehát egyetlen pozícióban van 1-es helyett 0. Három feltételnek kell teljesülnie a szomszédsághoz. A feltételekben példaként használt két bináris szám különbözik csak az i -edik bitben.

Az első feltétel azt írja elő, hogy a decimális indexek különbsége kettő hatványa kell, hogy legyen. Ha a két bináris számot kivonunk egymásból (a nagyobból a kisebbet), akkor eredményül olyan bináris számot kapunk, amely mindenhol 0, kivéve az i -edik bitet. Ennek decimálisan értéke 2^i . Például az $111_2 = ABC$ mintermnek három szomszédja van, az

$m_2^3 = ABC$, az $m_3^3 = A\bar{B}C$ és az $m_4^3 = AB\bar{C}$. Mind a három mintermmel való különbsége kettő hatványa, konkrétan 4, 2 és 1. Ez a feltétel önmagában nem elegendő, mivel például az $m_2^3 = ABC$ és az $m_1^3 = A\bar{B}C$ mintermek decimális indexének különbsége 1, de mégsem szomszédosak.

A második feltétel az indexek bináris súlyának különbséges 1 kell, hogy legyen. A bináris súly azt mondja meg, hogy a számot binárisan felírva hány darab 1-es van benne. Mivel a két bináris szám csak az i -edik bitben különbözik egymástól, ezért amelyikben 1-es van 0 helyett, annak a bináris súlya eggyel nagyobb. Például az $m_2^3 = ABC$ bináris súlya 3, a vele szomszédos $m_1^3 = A\bar{B}C$, $m_3^3 = A\bar{B}C$ és $m_4^3 = AB\bar{C}$ bináris súlya 2. Ez a feltétel sem elegendő önmagában, mivel például az $m_2^3 = ABC$ bináris súlya 1, az $m_3^3 = A\bar{B}C$ bináris súlya 2, azaz bináris súlyuk különbsége 1, de mégsem szomszédosak.

A harmadik feltétel szerint a nagyobb decimális indexnek nagyobb a bináris súlya. Amelyik számban 1-es van 0 helyett, annak decimális értéke 2^i értékkel nagyobb. Ez a feltétel sem elegendő önmagában, mivel például az $m_2^3 = ABC$ bináris súlya 2, az $m_1^3 = A\bar{B}C$ bináris súlya 0, azaz a nagyobb értékűnek nagyobb a bináris súlya, de mégsem szomszédosak.

A három feltétel együtt biztosítja számunkra azt, hogy a két minterm minimális legyen. Azaz összefoglalva két minterm szomszédos, ha a következő feltételek teljesülnek.

- A decimális indexük különbségének értéke 2 hatványa.
- A bináris súlyuk különbsége 1.
- A nagyobb indexű term súlya nagyobb.

A fenti feltételek megfelelően leírják, hogy két minterm mikor szomszédos, azaz mikor lehet őket összevonni. Ez viszont csak az első lépéshez elegendő, mivel az implikánsok nem csak egyetlen mintermet tartalmazhatnak. Az implikánsok összevonásánál a feltétel szerint az implikánsok által lefedett mintermek páronként szomszédosak. Ez csak abban az esetben fordulhat elő, ha két implikánsnak ugyanazokat a változókat tartalmazza, azaz ugyanazok a változók lettek belőlük kiegyszerűsítve.

Prímimplikánsok meghatározása

Ezek után gyakorlatilag ugyanaz a módszer használható, mint a Karnaugh tábla esetében. A kevesebb mintermet tartalmazó implikánsokat kell összevonni és nagyobbakat készíteni. Első lépésben az egy mintermet tartalmazó implikánsokat kell összegyűjteni, a Karnaugh táblában ez az egyes hurkokat jelenti. Ezeket egy táblázatba kell rendezni a bináris súlyuk alapján csoportosítva, mivel így a második feltétel ellenőrzésére a későbbiekben nem lesz szükség. A csoportok között aláhúzás szerepel a táblázatban a jobb láthatóság végett. A mintermek összevonása során a szomszédos csoportokban lévő összes mintermet párosítani kell és ellenőrizni, hogy a három korábban felírt feltétel teljesül-e. Amennyiben igen, akkor ez egy új implikánst eredményez, mely egy új táblázatban kerül összegyűjtésre. Ebben a táblázatban az implikánsok mellett fel kell jegyezni, hogy mennyi volt a decimális különbségük. A decimális különbség mutatja, hogy melyik változót lehetett kiegyszerűsíteni az összevonással, a decimális különbség 2-es alapú logaritmusa jelöli ki az elhagyható változó helyiértékét. Továbbá meg kell jelölni, hogy mely mintermeket lehetett bevonni valamely implikánsba, mivel ezek a mintermek nem lehetnek prímimplikánsok. Ilyen módon el lehet készíteni az összes kettő mintermet tartalmazó implikánst. Érdemes az újonnan kapott implikánsokat is az első mintermjük szerint bináris súlyuk alapján csoportosítani, ezt aláhúzással jelöljük a táblázatban, így csak a szomszédos csoportokban lévő implikánsokat kell megvizsgálni.

A Karnaugh tábla használatához hasonlóan itt is iteratívan addig kell új táblázatokat készíteni az előzőből, amíg keletkezik új implikáns. A ciklus végén azok az implikánsok lesznek a

prímimplikánsok, melyeket nem lehetett összevonni másik implikánssal, azaz nem lettek megjelölve az algoritmus során.

A példában legyen a minimalizálandó függvény ugyanaz, mint a Karnaugh táblás minimalizálás bemutatása során, azaz az $F^{n=4} = \sum_{i=0}^{15} (0,1,5,6,7,9,10,13,15)$. A könnyebb érthetőség miatt minden lépés után szerepel a Karnaugh tábla is. Első lépés a mintermek csoportosítása a bináris súlyuk szerint.

Minterm	Bináris érték	Bináris súly
0	0000	0
1	0001	1
5	0101	2
6	0110	
9	1001	
10	1010	
7	0111	3
13	1101	
15	1111	4

		CD			
		C			
AB		00	01	11	10
	00	1 ₀	1 ₁	0 ₃	0 ₂
	01	0 ₄	1 ₅	1 ₇	1 ₆
	11	0 ₁₂	1 ₁₃	1 ₁₅	0 ₁₄
A	10	0 ₈	1 ₉	0 ₁₁	1 ₁₀
		D			

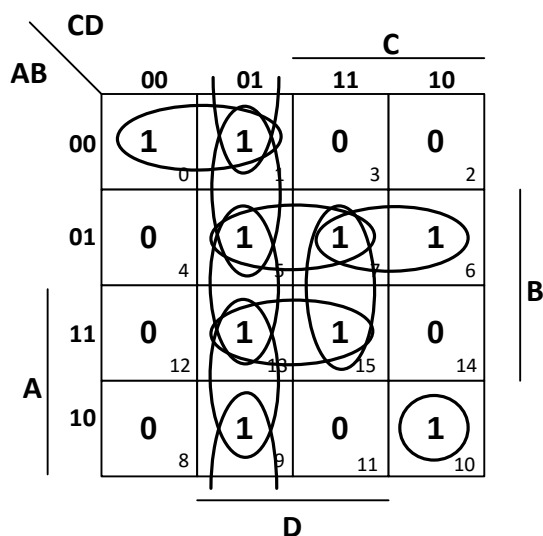
Következő lépésben a szomszédos csoportokban lévő mintermek kell összehasonlítani és amiket lehet, összevonni új implikánsá. Ahol mind a három, mintermekre vonatkozó szomszédossági feltétel teljesül, azok bekerülnek a következő táblázatba és az összevonható mintermek mellé „+” jel kerül. Például az 5 és 9 mintermre teljesülnek a feltételek, de például a 9 és 7 mintermekre nem teljesül a harmadik tulajdonság, azaz hogy a nagyobb bináris súlyú minterm decimális értéke nagyobb. Az összevonás eredményeképpen ugyanazok az implikánsok adódtak, mint a Karnaugh táblás minimalizálás során a kettes hurkok képzésénél és a 10-es minterm ugyanúgy nem vonható össze semmilyen másik mintermmel, azaz ő maga egy prímimplikáns.

Minterm	Bináris érték	Bináris súly
0+	0000	0
1+	0001	1
5+	0101	2
6+	0110	
9+	1001	
10	1010	
7+	0111	3

Implikáns	Decimális különbség
0, 1	(1)
1, 5	(4)
1, 9	(8)
5, 7	(2)
5, 13	(8)
6, 7	(1)
9, 13	(4)

13+	1101	
15+	1111	4

7, 15	(8)
13, 15	(2)



Ezek után következik az eredményül kapott implikánsok további összevonása. Ebben az esetben akkor vonhatóak össze az implikánsok, ha páronként teljesülnek a benne lévő mintermekre a szomszédossági feltétel és ebből következően a decimális különbségük megegyezik. Ez alapján össze lehet vonni az 1, 5 implikánst a 9, 13 implikánssal, ami a 1, 5, 9, 13 implikánst eredményezi ugyanúgy, mint az 1, 9 implikáns összevonása az 5, 13 implikánssal. A Karnaugh táblában is minden négyes hurkot két féle képen lehetett előállítani kettes hurkokból. Az összevonás során még egy új implikáns keletkezik, de annak decimális különbsége nem ugyanaz, mint a másiké, ezért ezeket a későbbiekben nem lehet összevonni.

Implikáns	Decimális különbség
0, 1	(1)
1, 5+	(4)
1, 9+	(8)
5, 7+	(2)
5, 13+	(8)
6, 7	(1)
9, 13+	(4)
7, 15+	(8)
13, 15+	(2)

Implikáns	Decimális különbség
1, 5, 9, 13	(4, 8)
5, 7, 13, 15	(2, 8)

		CD		C	
		00	01	11	10
AB	00	1	1	0	0
	01	0	1	1	1
	11	0	1	1	0
	10	0	1	0	1
		D		B	
		0	1	3	2
		4	5	7	6
		12	13	15	14
		8	9	11	10

A számjegyes minimalizálás során kapott prímmimplikánsok megegyeznek azokkal a prímmimplikánsokkal, amelyeket a Karnaugh táblás minimalizálás eredményezett.

A következő feladat a decimális indexekből a prímmimplikánsok algebrai alakját meghatározni. Meg kell vizsgálni, hogy mely pozíciókban egyforma az összes minterm algebrai alakja és ez adja meg az prímmimplikáns algebrai alakját. Ez alapján a következő képen kell meghatározni az előző példa prímmimplikánsainak algebrai alakjait.

Decimális alak	Algebrai alakok	Eredő algebrai alak
10	$ABCD$	$ABCD$
0, 1	$ABCD$ $ABCD$	ABC
6, 7	$ABCD$ $ABCD$	ABC
1, 5, 9, 13	$ABCD$ $ABCD$ $ABCD$ $ABCD$	CD
5, 7, 13, 15	$ABCD$ $ABCD$ $ABCD$ $ABCD$	BD

Egyszerűsített alak megadása

Az egyszerűsített alak meghatározásához elsősorban a prímmimplikáns tábla használható. Ebből kiolvashatók a lényeges prímmimplikánsok és szerencsés esetben ezek az összes mintermet tartalmazzák („lefedik”). Abban az esetben, ha a prímmimplikáns tábla alapján ránézésre nem állapítható meg, hogy mely prímmimplikánsokra van szükség, vagy többváltozós bonyolult függvényt kell minimalizálni, akkor segédfüggvényt használhatunk az eredmény meghatározásához.

A prímmimplikáns táblában minden prímmimplikánshoz tartozik egy sor és minden mintermhez tartozik egy oszlop. Ha egy prímmimplikáns tartalmaz egy mintermet, akkor az adott cellába „*”-ot kell írni. A megkülönböztetett mintermek oszlopaiban egyetlen „x” szerepel, azaz azok a lényeges prímmimplikánsok, amelyben van olyan minterm, melyek oszlopában egyetlen „x” van, ezeket „*”-al megjelöljük.

A $f(x) = \sum_{i=0}^{15} a_i x^i$ függvényhez öt prímisszám tartozik, azaz a prímisszám táblának 5 sora és 8 oszlopa lesz. A fentiek alapján kitölthető a táblázat és látható belőle, hogy 0 mintermet csak a 0, 1 prímisszám fed le, azaz ez lényeges prímisszám. Hasonlóan a 10, 11 és az 5, 7, 13, 15 prímisszámok is lényeges prímisszámok. A példában a lényeges prímisszámok lefedik az összes mintermet, azaz ránézésre látszik a megoldás.

Sor	Prímisszám	0	1	5	7	10	11	13	15
*	0, 1	x	x						
	1, 5		x	x					
*	10, 11					x	x		
	11, 15						x		x
*	5, 7, 13, 15			x	x			x	x

Olyan eset is előfordulhat, hogy nem a legegyszerűbb prímisszámok maradnak meg a kiválasztás után. A következő példában egy darab négy és négy darab kettő mintermet tartalmazó prímisszám van. A prímisszám táblából jól látszik, hogy az összes kettő mintermet tartalmazó prímisszám lényeges prímisszám jelöl és ezek lefedik az összes mintermet, azaz az egyszerűsített függvényalakban a négy mintermet tartalmazó prímisszámra nincs szükség.

Sor	Prímisszám	1	5	6	7	11	12	13	15
*	1, 5	x	x						
*	6, 7			x	x				
*	12, 13						x	x	
*	11, 15					x			x
	5, 7, 13, 15		x		x			x	x

A következő példa azt az esetet mutatja be, amikor nincs a rendszerben lényeges prímisszám, azaz ránézésre nem lehet egyértelműen eldönteni, hogy mely prímisszámokra van szükség a legegyszerűbb alak eléréséhez. Jelen esetben a prímisszám tábla minden oszlopában kettő darab „x” szerepel.

Sor	Prímisszám	1	3	7	9	13	15
	1, 3	x	x				
	1, 9	x			x		
	3, 7		x	x			
	7, 15			x			x
	13, 15					x	x
	9, 13				x	x	

Amikor ránézésre nem állapítható meg a legegyszerűbb alak, vagy amikor sok változó és prímisszám miatt áttekinthetetlen a táblázat, segédfüggvényt kell használni. A segédfüggvény gyakorlatilag logikai függvénye annak a feltételnek, hogy minden mintermet le kell fedni legalább egy prímisszámossal. Ennek eléréséhez minden prímisszámhoz be kell vezetni egy logikai változót, mely azt reprezentálja, hogy az szerepel-e az egyszerűsített függvényalakban. Ezeket jelöli az előző táblázatban az $a_1, a_3, a_7, a_9, a_{13}$ és a_{15} . Azt, hogy egy

minterm le van fedve, úgy lehet leírni, hogy az őt lefedni tudó prímiimplikánsokhoz tartozó logikai változókat VAGY kapcsolattal kell összefűzni. Azt, hogy minden minterm le van fedve, úgy lehet leírni, hogy a mintermekhez tartozó logikai kifejezéseket ÉS kapcsolattal kell összekapcsolni. Az eredményül kapott segédfüggvényt S-el jelölik.

A prímiimplikáns változóknak egy olyan kombinációját kell megtalálni, hogy a segédfüggvény értéke 1-et vegyen fel. Ehhez át kell alakítani a függvény diszjunktív alakba a szorzások (ÉS kapcsolatok) elvégzésével és meg kell határozni az összes prímiimplikánsát. Diszjunktív alak esetén a segédfüggvény akkor vesz fel 1 értéket, ha van olyan tagja, melynek értéke 1. Azaz ki lehet választani bármelyik tagot a segédfüggvényből, és ha az abban a tagban szereplő változókhoz tartozó prímiimplikánsok segítségével lesz megvalósítva a logikai függvény, akkor minden minterm le lesz fedve. Mivel a legegyszerűbb alak megtalálása a cél, ezért azt a tagot kell kiválasztani, amelyben a legkevesebb prímiimplikáns szerepel. Ha több olyan tag is van, melyek számossága minimális, akkor a választás közöttük tetszőleges.

Az előző példához tartozó segédfüggvény a következő képen néz ki $S = (a + b)(a + c)(c + d)(b + f)(e + f)(d + e)$. Ezt át lehet alakítani diszjunktív alakba a zárójelek felbontásával és egyszerűsíteni az elnyelési tulajdonság felhasználásával $S = (a + bc)(bc + cf + bd + df)(e + df) = (bc + acf + abd + adf)(e + df) = bce + bcdf + acf + acdf + abde + abdf + udef + adf = bce + adf + bcdf + acef + abde$. Az öt kapott prímiimplikáns öt különböző megoldást jelöl, ezek közül az a bce és az adf a legegyszerűbbek, mivel ezek tartalmazzák a legkevesebb prímiimplikánsot. A bce alapján a függvény megvalósításához a b , a c és az e prímiimplikánsokra van szükség, így a függvény egyszerűsített alakja $F = b + c + e = \overline{B}CD + \overline{A}CD + ABD$. Az adf alapján a függvény megvalósításához az a , a d és az f prímiimplikánsokra van szükség, így a függvény egyszerűsített alakja $F = a + d + f = \overline{A}BD + BCD + ACD$.

Példa

Legyen adott a következő logikai függvény $F^{n=4} = \sum_{i=0}^{2^4-1} (4,5,10,11,13,14,15)$. A mintermek alapján fel lehet írni a kiinduló táblázatot.

Minterm	Bináris érték	Bináris súly
4	0100	1
5	0101	2
10	1010	
11	1011	3
13	1101	
14	1110	
15	1111	4

Az első lépésben meg kell határozni a kettő mintermet tartalmazó implikánsokat.

Minterm	Bináris érték	Bináris súly	Implikáns	Decimális különbség
4+	0100	1	4, 5	(1)
5+	0101	2	5, 13	(8)
10+	1010		10, 11	(1)
11+	1011	3	10, 14	(4)
13+	1101		11, 15	(4)
14+	1110		13, 15	(2)
15+	1111	4	14, 15	(1)

Ezek után meghatározhatók a négy mintermet tartalmazó implikánsokat.

Implikáns	Decimális különbség	Implikáns	Decimális különbség
4, 5	(1)	10, 11, 14, 15	(1, 4)
5, 13	(8)		

10, 11+	(1)
10, 14+	(4)
11, 15+	(4)
13, 15	(2)
14, 15+	(1)

Mivel csak egyetlen darab négy mintermet tartalmazó implikáns van, ezért azt már nem lehet összevonni másikkal, azaz nincs nyolc mintermet tartalmazó implikáns. A prímiimplikáns tábla segítségével meghatározhatók a lényeges prímiimplikánsok.

Sor	Prímiimplikáns	4	5	10	11	13	14	15
*	4, 5	a	x	x				
	5, 13	b		x		x		
	13, 15	c				x		x
*	10, 11, 14, 15	d		x	x		X	x

A lényeges prímiimplikánsok nem fedik le az összes mintermet, ezért a segédfüggvényt segítségével kell meghatározni a szükséges prímiimplikánsokat. $S = a(a+b)cd(d+c+d) = ad(b+c) = abd + abc$. Azaz a két lényeges prímiimplikáns mellett tetszőleges, hogy a b vagy a c prímiimplikáns szerepel a megoldásban. Most legyen b a kiválasztott prímiimplikáns. Ekkor az eredő függvényhez meg kell határozni a prímiimplikánsok algebrai alakjait.

Decimális alak	Algebrai alakok	Eredő algebrai alak
4, 5	$AB\bar{C}\bar{D}$ $\bar{A}B\bar{C}\bar{D}$	$AB\bar{C}$
5, 13	$\bar{A}B\bar{C}D$ $AB\bar{C}D$	$B\bar{C}D$
10, 11, 14, 15	$AB\bar{C}\bar{D}$ $\bar{A}B\bar{C}D$ $AB\bar{C}D$ $ABCD$	AC

Ebből az eredő függvényalak $F = \bar{A}B\bar{C} + B\bar{C}D + AC$.

A következő példában legyen $F^{n=5} = \sum_{i=0}^{2^5-1} (0,1,6,7,14,15,17,21,22,23,24,30,31)$ egy ötváltozós logikai függvény. A mintermek alapján fel lehet írni a kiinduló táblázatot.

Minterm	Bináris érték	Bináris súly
0	00000	0
1	00001	1
6	00110	2
17	10001	
24	11000	
7	00111	3
14	01110	
21	10101	
22	10110	
15	01111	4
23	10111	
30	11110	
31	11111	5

Az első lépésben meg kell határozni a kettő mintermet tartalmazó implikánsokat.

A lényeges prímiplikánsok nem fedik le az összes mintermet, ezért a segédfüggvény segítségével kell meghatározni a szükséges prímiplikánsokat.

$$S = b(b+c)ffff(c+d)(d+e)f(e+f)aff = bf(c-d)(d+e)a = abf(cd+ce+d-d-e) = abf(ce+d) = abcef + abd f$$

. A két lehetséges megvalósításból az $abd f$ tartalmaz kevesebb prímiplikáns, ezért ez az optimális megoldás. Az eredő függvényhez meg kell határozni a prímiplikánsok algebrai alakjait.

Ebből az eredő függvényalak $F = ABC\bar{D}\bar{E} + \bar{A}BC\bar{D} + AB\bar{D}E + CD$.