Zalotay Péter

Digitális technika

Elektronikus jegyzet

Kandó Kálmán Villamosmérnöki Kar

Tartalomjegyzék

Bevezetés

Az elektronikus jegyzet a ***BMF Kandó Kálmán Villamosmérnöki Kar*** érvényes tantervében szereplő ***Digitális technika I***, tantárgy oktatási anyagát tartalmazza. A jegyzet három fő részben:

- ***A digitális technika elméleti alapjai,***
- ***A digitális hálózatok, és***
- ***Digitális integrált áramkörök, és alkalmazásuk***

fejezetekben tárgyalja a kötelező tananyagot. A tananyag elsajátítását segítik a tantermi foglalkozások során megoldott példák, és otthoni feladatok. A gyakorlati készség fejlesztését szolgálják laboratóriumi gyakorlatok. Mindezekhez bőséges oktatási segédlet áll a nappali, a levelező, és a távoktatásos hallgatók részére.

A ***digitális technika*** módszereivel az ***információ leképzés***, ***műveletvégzés*** és az eredmények továbbítása kétértékű elemi információk (bitek) sorozatával, digitális szavakkal történik. A különböző műveletvégzések egyszerű logikai döntések sorozatára

vezethetők vissza. Ugyancsak logikai műveleteket kell végezni, pl. két - különböző mennyiség értékét hordozó - információ közötti viszony (kisebb, nagyobb, egyenlő) megállapításához.

Mielőtt a digitális technika alapjairól írnánk, röviden ismerkedjünk meg – a teljesség igénye nélkül – az e - technikát megalapozó legjelentősebb személyek munkásságával.

George Boole (1815-1864) angol matematikus foglakozott legelőször a formális logika algebrai szintű leírásával és alkotta meg a róla elnevezett algebrát, melyet 1847-ben a " The Mathematical Analiysis of Logic " című könyvében tett közzé.

C. Shannon mérnök-matematikus 1938 -ban megjelent 'Switching Theory' című könyvében adaptálta először G. Boole algebráját kétállapotú kapcsolóelemeket tartalmazó logikai rendszerek leírására. Az információelmélet megalapítása is nevéhez fűződik, az információ alapegységét is tiszteletére róla nevezték el

Azóta hihetetlen mértékű fejlődés következett be a technika és ezen belül is a logikai rendszerek fejlődésében és alkalmazásában. Ez a fejlődés mind az elmélet, a rendszertechnika mind pedig a technológia területén igen gyors volt és természetesen ma is még az. A technológia fejlődésén természetesen itt elsősorban az áramköri elemek és az ehhez kapcsolódó logikai illetve áramköri rendszerek szerelésének automatizálásra lehet gondolni. Érdekes megfigyelni - véleményem szerint a technika fejlődésében egyedülálló módon - hogy voltak időszakok amikor a technológia fejlődése - konkrétan a nagy bonyolultságú integrált áramkörök, a mikroprocesszorok megjelenése - készületlenül érte az elméletet, szinte lehagyva azt.

A következő felsorolás teljesen önkényes, de mindenképpen olyan tudománytörténeti neveket tartalmaz akik igen nagy mértékben elősegítették a logikai rendszerek elméletének kidolgozását, fejlődését,

Evarist Galois (1812-1832) Francai matematikus a modern algebra egyik ágának megalapítója. Az általa létrehozott és róla elnevezett csoportelmélet adja a kódolás elmélet, a kriptográfia elméleti hátterét. Rövid élete alatt hozta létre ezt a nem éppen könnyen elsajátítható elméletet, még egyetemista korában párbajban meghalt.

Wilkes angol matematikus aki 1954 es években kifejlesztette a mikro-programozás elméletét, amelyet a technológia akkori szintjén még igen költséges lett volna alkalmazni. Ez az elmélet többek között a számítógépek központi vezérlőegységének tervezéshez adott univerzális megoldást. Első alkalmazásai között az igen népszerű IBM 360 -as számítógép is szerepelt.

1964-65 években Mealey és Moore mérnökök a logikai rendszerek tervezésének egy olyan zárt jól alkalmazható elméletét adták meg, mely a kor eszközbázisának megfelelő alkalmazását tette lehetővé.

Az 1971-es évre tehető az integrált áramköri gyártástechnológia olyan mértékű fejlődése, hogy lehetőséggé vált a számítógépek központi egységének megvalósítása egy vagy több tokban, vagyis megjelent a mikroprocesszor. Azóta a fejlődés még inkább felgyorsult és szinte nincs az iparnak, a szórakoztató-iparnak, a kereskedelemnek, a mezőgazdaságnak, a szolgáltatásoknak olyan területe, ahol a nagy integráltságú és olcsó digitális rendszerek ne terjedtek volna el. Kis túlzással azt mondhatnánk, hogy az utolsó egy két évtized a digitális technika korszaka volt és talán még marad is. Az integrált áramkörök gyártástechnológiájának fejlődését igen jól mutatja az , hogy az 1972-es évek közkedvelt I8080 típusú mikroprocesszora még csak megközelítően 4700 tranzisztort tartalmazott, míg ma a kereskedelemben lehet kapni olyan Pentium alapú mikroprocesszort és egyéb rendszertechnikai elemeket tartalmazó chipet mely 150 millió tranzisztorból épül fel

Természetesen nem csak mikroprocesszorokat fejlesztettek ki, de más univerzálisan, vagy nagy sorozatban használható áramköri készletek is kialakultak:

- ***memóriák***
- ***programozhat logikai elemek: FPGA, stb.***
- ***berendezés orientált integrált áramkörök***
- ***céláramkörök, pl. Quarz órák***

Az integráltság mértékének növekedésével egyre több funkció került egy tokba (chip-be), amely jelentősen megnövelte a kivezetések számát is. Ezeknek a nyomtatott

áramköri lemezre való beültetésére a hagyományos technológia nem volt alkalmas, ezért kifejlesztették a felületszerelési technológiákat (angolul Surface Mount Technology = SMT) és alkatrészeket (angolul Surface Mountage Devices) SMD.

Az egy chipben leintegrált logikai funkciók olyan bonyolultakká váltak, hogy tesztelésükre már a hagyományos módon nem volt lehetőség, ezért ki kellett fejleszteni új megoldásokat erre a feladatra, és ezek a ma oly közkedvelt szimulációs programok illetve hardware leíró nyelvek (VHDL).

Nagyon kevés műszaki szakterületet lehet találni, amelynek csak megközelítően is akkora irodalma volna mint a digitális technikának illetve rendszereknek. Ugyanakkor és ez talán ellentmondásnak tűnik, hogy ritka az olyan szakterület is amelyben olyan rövid idő alatt lehet olyan tudásra szert tenni , mellyel már egész komoly logikai rendszerek építhetők fel. Az ellentmondást az oldja fel, hogy ma már nem elegendő ha egy rendszer működik, ez csak egy alapkövetelmény, de annak számos esetben igen nagy megbízhatósággal, könnyű szervizelhetőséggel, versenyképes áron kell megvalósulnia. És az ilyen "hiba tűrő" rendszerek tervezése és szervizelése nagy tudást igényel.

1. A DIGITÁLIS TECHNIKA ELMÉLETI ALAPJAI

1.1. Logikai alapismeretek

Mint ahogyan azt a bevezetőben is említettük, a ***digitális technika*** a ***műszaki***, ***technikai*** folyamatok ***megvalósítására*** alkalmas ***berendezések***, automaták tervezéséhez szükséges ***elmélettel***, ***módszerekkel***, és ***áramkörökkel*** foglalkozik.

A tervezendő készülékek, berendezések be-, és kimeneteinek jelei (logikai változói) csak ***két értéket*** vehetnek fel, és a ***döntések*** a formális ***logikában*** használt ***műveleteken*** alapulnak. A változók teljes halmazt alkotnak, amelyet eseménytérnek is nevezhetünk.

A következőkben először összefoglaljuk röviden a használt halmazelméleti alapfogalmakat. Majd tárgyaljuk a logikai algebra rendszerét, valamint alkalmazási lehetőségeit, módszereit.

1.2. Halmazelméleti alapfogalmak

Halmazon valamilyen ***közös*** tulajdonsággal rendelkező dolgok ***összességét*** értjük. A halmazhoz tartozó "dolgok összességét" a halmaz ***elemeinek*** nevezik. Az adott tulajdonságokkal nem rendelkező dolgok összessége alkotja a ***komplemens*** vagy kiegészítő halmazt.

A halmazok lehetnek ***végesek*** vagy ***végtelenek*** a halmazt alkotó elemek számától függően. Két speciális halmazt is definiálnak: üres halmaz melynek egyetlen eleme sincs, és a teljes vagy ***univerzális*** halmazt, amelyet valamely halmaz és ennek ***komplemens*** - e alkot.

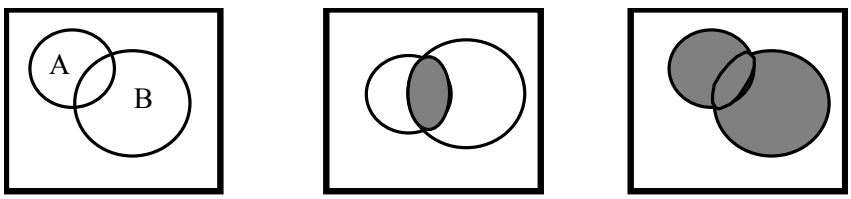
Egy halmaz általában további részekre úgy nevezett ***részhalmazokra*** is oszthatunk, mely úgy jön létre, hogy az adott halmazhoz még további szűkítő feltételt is rendelünk. Például vegyük egyszerűség kedvéért a természetes számok halmazát. A természetes számok részhalmazai lehetnek pl. a prímszámok, a 2-vel vagy a 3-al osztható számok stb.

Azon részhalmazt mely minden eleme része két vagy több halmaznak, azt a két halmaz ***közös részének*** (***metszet***) vagy latin kifejezéssel élve a két halmaz ***konjunkció*** - jának mondjuk.

A természetes számok közül tartalmazza az **A** halmaz a 2-vel, a **B** halmazt pedig a 3-mal osztható számokat. Azok a természetes számok melyek 2-vel és 3-mal is oszthatók a két halmaz ***közös*** részét más szóval ***metszetét*** képezik. Általánosan tehát az A halmaz elemei 2i ahol i $\subset$ [1, ∞], a B halmazé 3j ahol j $\subset$ [1, ∞], és így a közös rész halmazát a 6k ahol k $\subset$ [1, ∞] számok képezik. A közös rész jelölésére a halmazelméletben a **Λ**, vagy $\cap$ jelet használják.**(A Λ B,** vagy **A $\cap$ B)**

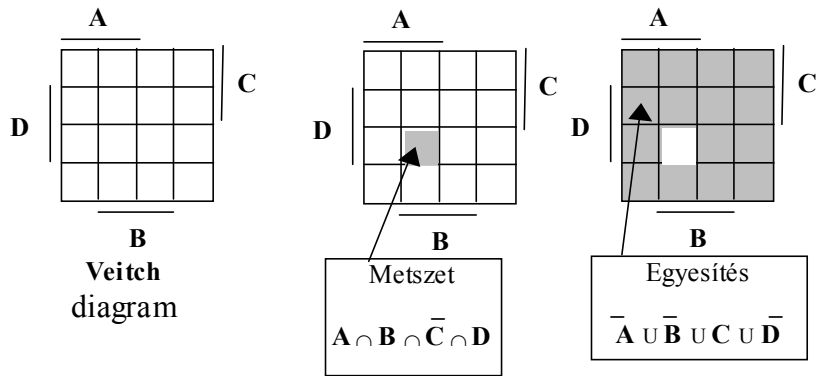
Azon elemekből felépülő halmazt mely tartalmazza mind az **A** mind pedig a **B** (vagy esetleg több halmaz) elemeit a két halmaz ***egyesített*** halmazának vagy ***uniójának*** nevezzük. Latin szóval ez a műveletet a diszjunkció. Előbbi példánknál maradva az egyesített halmaz elmei 6i, 6i-2, 6i-3, 6i-4 (i=1,2,3...). Az unió jelölésére az U, vagy a V jelöléseket használják. (**A U B** vagy **A V B**)

A halmazok és a rajtuk értelmezett műveletek jól szemléltethetők (a J.Venn és Veitch matematikusról elnevezett) ***diagramokkal*** is. A ***teljes halmazt*** egy ***négyszöggel***, míg a ***részhalmazokat*** egy ***zárt alakzattal*** célszerűen egy ***körrel*** – a ***Venn*** diagramban 1.ábra - vagy ugyancsak ***négyszöggel*** jelölik a 2.ábra szerinti ***Veitch*** diagramban.



Venn diagram | részhalmazok metszete | részhalmazok egyesítése

1. ábra



a. b. c.

2. ábra

A ***Veitch*** diagramban minden változó ***IGAZ*** értékéhez a teljes halmaz (esemény-tér) ***fele***, míg a ***másik*** térfél ugyanezen változó ***tagadott*** értékéhez tartozik. (Az algebrai leírásnál a változó fölé-húzásával jelöljük a tagadást). Az ábra négyváltozós halmazt ábrázol. A peremezésnél vonalak jelzik, hogy az egyes változók melyik térfélen IGAZ értékűek. A 2.b. ábrán a metszésnek (ÉS művelet) azt a változatát szemlélteti, amelyik mindegyik változó valamelyik értékének közös területe. Ez metszi ki a legkisebb elemi területet, ezért nevezik ezt ***minterm*** - nek. A 2.c ábrán az összes változó valamely értékeihez tartozó együttes terület. Az egyesített terület a legnagyobb részterület, amelyet ***maxterm*** -nek neveznek. Mind a két kitüntetett területből 2^n –en darab van, ahol n a változók száma.

1.3. A logikai algebra

A ***logikai*** algebra a ***Boole*** algebra alapjaira épül. Kiegészítésekkel a digitális rendszerek tervezésére, elemzésére alkalmas algebrává fejlődött.

A továbbiakban összefoglaljuk a logikai algebra alapjait. A logikai áramkörök később sorra kerülő ismertetésénél, valamint azok működésének megértéséhez az algebrai alapok biztos ismerete elengedhetetlen.

⇒ *Logikai változók, és értékük*

A ***logikai algebra*** csak ***kétértékű logikai változók*** halmazára értelmezett.

A ***logikai*** változók két csoportba oszthatók, úgymint

független-, és

függő változókra.

Mindkét csoport tagjait a latin ABC nagy betűivel (A, B, C . . . X, Y, Z) jelöljük. Általában az ABC első felébe eső betűkkel a független, az utolsó betűk valamelyikével, pedig a függő változókat jelöljük.

A változók két logikai értéke az ***IGAZ***, ill. a ***HAMIS*** érték. Ezeket **1**-el, ill. **0**-val is jelölhetjük (IGAZ: 1; HAMIS: 0).

1.4. A logikai algebra axiómái

Az ***axiómák*** olyan előre rögzített kikötések, ***alapállítások***, amelyek az algebrai rendszerben mindig érvényesek, viszont nem igazolhatók. Ezen állítások meghatározzák a halmaz ***elemeit***, a ***műveleteket***, azok ***tulajdonságait***. A ***tételek***, viszont az axiómák segítségével bizonyíthatók.

1. Az algebra ***kétértékű*** elemek halmazára értelmezett.

2. A halmaz minden elemének létezik a ***komplemens*** -e is, amely ugyancsak eleme a halmaznak, tehát ***teljes*** halmazt alkotnak.

3. Az elemek között végezhető ***műveletek***

- a ***konjunkció*** (logikai ÉS), illetve
- a ***diszjunkció*** (logikai VAGY).

4. A logikai műveletek ***tulajdonságai***:

- ***kommutatív*** –ak (a tényezők felcserélhetők),
- ***asszociatív*** – ak (a tényezők csoportosíthatók),
- ***disztributív*** – ak (a két művelet elvégzésének sorrendje felcserélhető).

5. A halmaz ***kitüntetett*** elemei az

- ***egység*** elem (értéke a halmazon belül mindig IGAZ), és a
- ***null*** elem (értéke a halmazon belül mindig HAMIS).

A ***logikai algebra*** a felsorolt axiómákra épül. A logikai feladatok technikai megvalósításához a halmaz egy elemének komplemenést képező művelet is szükséges. Ezért a műveletek között a logikai ***TAGADÁS*** (más szóhasználattal ***nem***, ***negáció***, ***invertálás***) is szerepel.

1.5. Logikai műveletek

A logikai algebra a következő logikai műveleteket alkalmazza. A változók logikai műveletekkel összekapcsolva alkotnak egy **logikai kifejezést**.

- ***ÉS*** **(**konjunkció**, AND)** - logikai szorzás;
- ***VAGY*** **(**diszjunkció**, OR)** - logikai összeadás;
- ***NEM*** **(**negáció**, invertálás, NOT)** - logikai tagadás.

A felsorolt műveletek közül az ***ÉS***, ill. a ***VAGY*** művelet ***két-***, vagy ***többváltozós***. Ez azt jelenti, hogy a változók legalább ***két eleme***, vagy ***csoportja*** között értelmezett logikai kapcsolatot határoz meg. A ***tagadás egy változós*** művelet, amely a ***változók***, vagy ***változócsoportok*** bármelyikére vonatkozhat.

A továbbiakban ismerkedjünk meg az egyes logikai műveletek definíciójával, és tulajdonságával.

⇒ *Az ÉS (AND) művelet*

A logikai változókkal végzett ***ÉS*** művelet ***eredménye akkor*** és ***csak akkor IGAZ*****,** ha ***mindegyik*** változó értéke egyidejűleg ***IGAZ***. A logikai algebrában az ÉS kapcsolatot szorzással jelöljük (logikai szorzás).

(Megjegyzés: a logikai szorzás jelet - akár csak az Euklideszi algebrában - nem szokás kitenni, így a továbbiakban mi is eltekintünk ettől).

Az

$$\mathbf{AB = K}$$

logikai függvényben az **A** és a **B** a ***független változók***, a **K** pedig a ***függő változó***, vagy eredmény. Jelentése pedig az, hogy a K akkor IGAZ, ha egyidejűleg az A és a B is IGAZ.

Fontos: a példában szereplő független változók egyedi változók, vagy egy-egy másik logikai függvény megoldásának eredményei is lehetnek

1.Példa:

Ahhoz, hogy egy szobában a lámpa világítson, alapvetően két feltételnek kell teljesülni:

- legyen hálózati feszültség;

- a kapcsoló bekapcsolt állapotban legyen.

Szóban megfogalmazva: **ha** van hálózati feszültség **és** a kapcsoló bekapcsolt, ***akkor*** a lámpa világít. (Az egyéb követelmények teljesülését, hogy az áramkör elemei jók feltételezzük.) Ebben az egyszerű technikai példában a ***hálózati feszültség*** és a ***kapcsoló állapota*** a ***független***-, a lámpa működése, pedig a ***függő változó***. Mindhárom tényező kétértékű.

⇒ *A VAGY (OR) művelet*

A logikai ***változókkal*** végzett ***VAGY*** művelet eredménye akkor ***IGAZ***, ha a független változók közül ***legalább az egyik*** IGAZ.

Algebrai formában ezt a független változók összegeként írjuk le (logikai összeadás). Az

$$\mathbf{A + B = K}$$

alakú algebrai egyenlőségben a K eredmény akkor IGAZ, ha vagy az A, vagy a B, vagy mindkettő IGAZ.

2.Példa:

Erre a logikai kapcsolatra ismert technikai példa egy gépkocsi ***irányjelzőjének*** működését ***ellenőrző lámpa***. A vezető előtt a műszerfalon levő lámpa ***világít***, ha a külső irányjelzők közül ***vagy*** a ***jobb*** oldali, vagy a ***bal*** oldali jelzőlámpacsoport világít. Azt az állítást, hogy jobb oldali jelzés van, jelölje **J** és azt, hogy bal oldali a jelzés, pedig **B**. Az eredményt, hogy a belső ellenőrző lámpa világít, jelöljük **L**-lel. A működést leíró logikai egyenlőség:

$$\mathbf{B} + \mathbf{J} = \mathbf{L}$$

alakú lesz.

⇒ *A TAGADÁS (INVERS) művelete*

A logikai tagadást ***egyetlen változón***, vagy ***csoporton*** végrehajtott műveletként értelmezzük. Jelentése, pedig az, hogy ***ha*** a ***változó IGAZ, akkor*** a ***tagadottja HAMIS*** és fordítva. Algebrai leírásban a tagadást a változó jele fölé húzott vonallal jelöljük. Ezek szerint a

$$\mathbf{K} = \overline{\mathbf{A}}$$

egyenlőség azt jelenti, hogy a **K** akkor IGAZ, ha az **A** HAMIS. (Szóban A nem - nek, A felülvonásnak vagy A tagadottnak mondjuk.)

Az

$$\overline{\mathbf{A} * \mathbf{B}} = \mathbf{K}$$

összefüggés azt írja le, hogy az eredmény (K) csak akkor igaz, ha az ***A*B*** logikai ***ÉS*** művelet eredménye ***HAMIS*** értéket ad.

3.Példa:

A tagadás műveletének előzőek szerinti értelmezése alapján abban a példában, amelyet az ÉS művelet magyarázatára hoztunk az $\overline{\mathbf{A}}$ (A nem) azt jelenti, hogy ***nincs hálózati feszültség***, ill. a $\overline{\mathbf{B}}$ (B nem) jelenti azt, hogy a kapcsoló ***nincs bekapcsolva***. Az eredmény tagadása ($\overline{\mathbf{K}}$) azt fejezi ki, hogy a lámpa ***nem világít***. Az előzőek alapján a gépkocsi irányjelzését ellenőrző lámpa működését leíró összefüggésben is értelmezhetjük a $\overline{\mathbf{J}}$-t (jobb oldali jelzés nincs), a $\overline{\mathbf{B}}$-t (bal oldali jelzés nincs) és az $\overline{\mathbf{L}}$ - t (ellenőrző lámpa nem világít) jelölések technikai tartalmát.

1.6. A logikai műveletek tulajdonságai

A következőkben a logikai **ÉS**, valamint logikai **VAGY** műveletek tulajdonságait elemezzük.

⇒ *Kommutativitás (tényezők felcserélhetősége)*

A leírt szemléltető példákat vegyük ismét elő. Azt állítottuk, hogy ha van hálózati feszültség, és a kapcsoló bekapcsolt, akkor a lámpa világít. Az eredmény változatlan, ha az állítások sorrendjét ***felcseréljük***, vagyis ha a kapcsoló be van kapcsolva és van hálózati feszültség, akkor világít a lámpa. Ez a látszólagos szójáték arra utal, - ami általánosan igaz - hogy az ***ÉS*** műveletekben a ***változók sorrendje felcserélhető***, amely algebrai formában az

$$\mathbf{AB} = \mathbf{BA}$$

azonossággal írható le.

Az előzőekhez hasonlóan meggyőződhetünk arról is, hogy a VAGY műveletekben is felcserélhető -ek az egyes állítások. Érvényes a

$$\mathbf{J} + \mathbf{B} = \mathbf{B} + \mathbf{J}$$

azonosság.

Tehát mindkét többváltozós logikai művelet ***kommutatív***.

⇒ *Asszociativitás (a tényezők csoportosíthatósága)*

A két logikai művelet további tulajdonsága a műveleti tényezők ***csoportosíthatósága*** is, vagyis az ***asszociativitás***. Algebrai alakban az

$$\mathbf{ABC} = \mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C} = \mathbf{B}(\mathbf{AC})$$

ill. az

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

azonosságok írják le az asszociatív tulajdonságot. A zárójel - a matematikai algebrához hasonlóan - a műveletvégzés sorrendjét írja elő. Eszerint a háromváltozós ÉS, ill. VAGY műveletet úgy is elvégezhetjük, hogy előbb csak két változóval képezzük az ÉS, ill. a VAGY kapcsolatot, majd annak eredménye és a harmadik változó között hajtjuk végre az előírt műveletet.

⇒ *Disztributivitás (a műveletek azonos értékűek)*

A harmadik jelentős tulajdonság, hogy a logikai ÉS, valamint a logikai VAGY ***azonos értékű*** művelet. Mindkettő disztributív a másikra nézve. Algebrai formában ez a következőképpen irható le:

$$\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$$

$$\mathbf{A} + \mathbf{BC} = (\mathbf{A} + \mathbf{B})(\mathbf{A} + \mathbf{C})$$

Az első azonosság alakilag megegyezik a matematikai algebra műveletvégzés szabályával. A második azonosság csak a logikai algebrában érvényes. Kifejezi azt, hogy egy logikai szorzat (ÉS kapcsolat) és egy állítás VAGY kapcsolata úgy is képezhető, hogy ***először*** képezzük a ***VAGY műveletet a szorzat*** tényezőivel és az így kapott eredményekkel hajtjuk végre az ***ÉS*** műveletet.

A logikai műveletek megismert tulajdonságai segítségével a logikai kifejezések algebrai átalakítása hajtható végre, és így lehetőség van a legegyszerűbb alakú kifejezés megkeresésére. Ezt a későbbiekben még részletesebben fogjuk tárgyalni.

1.7. A logikai algebra tételei

A továbbiakban felsoroljuk a fontosabb **tételeket**, azok részletes bizonyítása nélkül.

⇒ ***A kitüntetett elemekkel végzett műveletek:***

$$1*1 = 1 \qquad 0*0 = 0$$

$$1*A = A \qquad 0*A = 0$$

$$1+1 = 1 \qquad 0+0 = 0$$

$$1+A = 1 \qquad 0+A = A$$

⇒ ***Az azonos változókkal végzett műveletek:***

$$A*A = A \qquad A * \overline{A} = 0$$

$$A+A = A \qquad A + \overline{A} = 1$$

Fontos: hogy az A-val jelzett logikai változó nem csak egy változó, hanem egy logikai műveletcsoport eredményét is jelentheti.

⇒ ***A logikai tagadásra vonatkozó tételek:***

$$\overline{\overline{A}} = A \qquad \overline{\overline{\overline{A}}} = \overline{A}$$

Általánosan: a ***páros*** számú tagadás ***nem*** változtatja meg az értéket, míg a ***páratlan*** számú tagadás azt az ***ellenkezőjére*** változtatja.

⇒ ***Logikai kifejezés tagadása:***

$$\overline{(A+B)} = \overline{A} * \overline{B} \qquad \overline{A*B} = \overline{A} + \overline{B}$$

Az előző két tétel az un. ***De Morgan - tételek***, amelyek általánosan azt fogalmazzák meg, hogy egy logikai kifejezés tagadása úgy is elvégezhet, hogy az egyes változókat

tagadjuk, és a logikai műveleteket felcseréljük (VAGY helyett ÉS, ill. ÉS helyett VAGY műveletet végzünk).

⇒ ***Általános tételek:***

$$\mathbf{A(A+B)=A} \quad \mathbf{A+AB=A}$$

E két tétel a műveletek disztributív tulajdonsága és a már felsorolt tételek segítségével a következőképpen bizonyítható:

$$\mathbf{A(A+B)=AA+AB=A(1+B)=A}$$

$$\mathbf{A+AB=(A+A)(A+B)=A(A+B)=A}$$

⇒ ***További általános tételek***

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

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

$$\mathbf{AB+\overline{A}B=B}$$

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

$$\mathbf{AB+BC+\overline{A}C=AB+\overline{A}C}$$

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

A legutóbb felsorolt tételek is bizonyíthatók az alaptulajdonságok segítségével.

1.8. Algebrai kifejezések

A továbbiakban ismertetünk néhány módszert, amelyeket az algebrai kifejezések átalakításánál gyakran használunk.

⇒ ***Az algebrai kifejezés bővítése.***

Egy logikai szorzat értéke nem változik, ha a kifejezés és az **1**-el logikai szorzatát képezzük (ÉS).

$$\mathbf{AB = AB * 1}$$

Az 1-et, pedig felírhatjuk, pl. $(\mathbf{C} + \overline{\mathbf{C}})$ alakban. Tehát:

$$\mathbf{AB} = \mathbf{AB}(\mathbf{C} + \overline{\mathbf{C}}) = \mathbf{ABC} + \mathbf{AB}\overline{\mathbf{C}}$$

Egy logikai összeadás nem fog megváltozni, ha a kifejezés és a 0 logikai összegét képezzük (VAGY):

$$\mathbf{D} + \mathbf{E} = \mathbf{D} + \mathbf{E} + \mathbf{0}$$

A 0-t kifejezhetjük $\mathbf{F} * \overline{\mathbf{F}}$ alakban. A bővítést végrehajtva az

$$\mathbf{D} + \mathbf{E} = (\mathbf{D} + \mathbf{E}) + \mathbf{F} * \overline{\mathbf{F}} = (\mathbf{D} + \mathbf{E} + \mathbf{F})(\mathbf{D} + \mathbf{E} + \overline{\mathbf{F}})$$

azonosságot kapjuk.

Ennél a bővítésnél felhasználtuk a disztributivitást leíró egyik algebrai összefüggést, mely szerint

$$\mathbf{A} + \mathbf{BC} = (\mathbf{A} + \mathbf{B})(\mathbf{A} + \mathbf{C})$$

Az előzőben ismertetett bővítési szabály megfordítva egyszerűsítésre is felhasználható.

4.Példa:

Igazoljuk a tételek között felsorolt

$$\mathbf{AB} + \mathbf{BC} + \overline{\mathbf{A}}\mathbf{C} = \mathbf{AB} + \overline{\mathbf{A}}\mathbf{C}$$

azonosságot!

Első lépésként a baloldal mindhárom tagját kibővítjük úgy, hogy szerepeljen bennük mindegyik független változó (A,B,C).

$$\mathbf{AB}(\mathbf{C} + \overline{\mathbf{C}}) + \mathbf{BC}(\mathbf{A} + \overline{\mathbf{A}}) + \overline{\mathbf{A}}\mathbf{C}(\mathbf{B} + \overline{\mathbf{B}}) =$$
$$= \underline{\mathbf{ABC}} + \mathbf{AB}\overline{\mathbf{C}} + \underline{\mathbf{ABC}} + \underline{\underline{\overline{\mathbf{A}}\mathbf{BC}}} + \underline{\underline{\overline{\mathbf{A}}\mathbf{BC}}} + \overline{\mathbf{A}}\overline{\mathbf{B}}\mathbf{C}$$

Az így kapott hat szorzatot tartalmazó kifejezésben kettő - kettő azonos. Ezek közül egy - egy elhagyható az A + A = A tétel analógiájára (pl. ABC +…. + ABC = ABC). Ezeket jelöltük egyszeres, illetve kettős aláhúzással.

Második lépésként a bővítés fordítottját végezzük, vagyis ahol lehet az azonos tényezőket, kiemeljük.

$$\mathbf{ABC + AB\overline{C} + \overline{A}BC + \overline{A}\overline{B}C = AB(C + \overline{C}) + \overline{A}C(B + \overline{B}) = AB + \overline{A}C}$$

A zárójelekben levő kifejezések 1 értékűek.

Ezzel igazoltuk az eredeti azonosságot.

Algebrai kifejezés tagadása (a De Morgan - tételek alkalmazása).

$$\overline{\mathbf{AB\overline{C} + \overline{A}BC + \overline{A}\overline{B}C}} = \mathbf{(\overline{A}\overline{B}\overline{C})\ (\overline{\overline{A}BC})\ (\overline{\overline{A}\overline{B}C}) =}$$
$$\mathbf{= (\overline{\overline{A}} + B + \overline{C})\ (A + \overline{B} + \overline{C})\ (A + B + \overline{C}) =}$$
$$\mathbf{(\underline{\overline{\overline{A}}A + AB + A\overline{C} + \overline{A}\overline{B} + B\overline{B} + \overline{B}\overline{C} + \overline{A}\overline{C} + B\overline{C} + \overline{C}\overline{C}})\ (A + B + \overline{C}) =}$$

Az átalakításnál először a De Morgan - tételt használtuk (első és második sor). A következő lépésként az első két zárójeles kifejezés logikai szorzatát (ÉS művelet) képeztük (az eredmény aláhúzva).

Az <u>aláhúzott</u> részt célszerű tovább egyszerűsíteni az $\mathbf{\overline{A}A = 0}$, és a $\mathbf{B\overline{B} = 0}$ tényezők elhagyásával, illetve a $\mathbf{\overline{C}\overline{C} = \overline{C}}$ helyettesítéssel. Majd tovább is egyszerűsíthető a $\mathbf{\overline{C}}$ kiemelésével.

$$\mathbf{(\underline{AB + A\overline{C} + \overline{A}\overline{B} + \overline{B}\overline{C} + \overline{A}\overline{C} + B\overline{C} + \overline{C}}) = AB + \overline{\overline{A}\overline{B}} + \overline{C}(A + \overline{B} + \overline{A} + B + 1) =}$$
$$\mathbf{= AB + \overline{\overline{A}\overline{B}} + \overline{C}}$$

A zárójelben levő kifejezés azonosan 1, mert a logikai összeadás egyik tagja 1. Térjünk vissza az eredeti kifejezéshez, amelynél a zárójelbe tett kifejezések ”összeszorzása”, majd a lehetséges további átalakítás után (pl. az aláhúzott kifejezések értéke 0 stb.) kapjuk meg a végeredményt.

$$= (\mathbf{AB} + \overline{\mathbf{A}}\overline{\mathbf{B}} + \overline{\mathbf{C}})(\mathbf{A} + \mathbf{B} + \overline{\mathbf{C}}) = \mathbf{ABA} + \underline{\overline{\mathbf{A}}\overline{\mathbf{B}}\mathbf{A}} + \mathbf{A}\overline{\mathbf{C}} + \mathbf{ABB} + \underline{\overline{\mathbf{A}}\overline{\mathbf{B}}\mathbf{B}} + \overline{\mathbf{C}}\mathbf{B} +$$
$$+ \mathbf{AB}\overline{\mathbf{C}} + \overline{\mathbf{A}}\overline{\mathbf{B}}\overline{\mathbf{C}} + \underline{\overline{\mathbf{C}}\overline{\mathbf{C}}} = \mathbf{AB} + \mathbf{0} + \mathbf{A}\overline{\mathbf{C}} + \mathbf{AB} + \mathbf{0} + \overline{\mathbf{C}}\mathbf{B} + \mathbf{AB}\overline{\mathbf{C}} + \overline{\mathbf{A}}\overline{\mathbf{B}}\overline{\mathbf{C}} + \mathbf{0} =$$
$$= \mathbf{AB}(\mathbf{1} + \mathbf{1} + \overline{\mathbf{C}}) + \overline{\mathbf{C}}(\mathbf{A} + \mathbf{B} + \overline{\mathbf{A}}\overline{\mathbf{B}}) = \mathbf{AB} + \overline{\mathbf{C}}$$

***5.Példa**:*

Igazoljuk a

$$\overline{\overline{\mathbf{D}}\overline{\mathbf{F}} + \mathbf{E}\overline{\mathbf{F}}} = \mathbf{F} + \mathbf{D}\overline{\mathbf{E}}$$

azonosságot!

Első megoldás:

$$\overline{\overline{\mathbf{D}}\overline{\mathbf{F}} + \mathbf{E}\overline{\mathbf{F}}} = \overline{(\overline{\mathbf{D}} + \mathbf{E})\overline{\mathbf{F}}} = \overline{(\overline{\mathbf{D}} + \mathbf{E})} + \mathbf{F} = \mathbf{D}\overline{\mathbf{E}} + \mathbf{F}$$

Második megoldás**:**

$$\overline{\overline{\mathbf{D}}\overline{\mathbf{F}} + \mathbf{E}\overline{\mathbf{F}}} = (\overline{\overline{\mathbf{D}}\overline{\mathbf{F}}})(\overline{\mathbf{E}\overline{\mathbf{F}}}) = (\mathbf{D} + \mathbf{F})(\overline{\mathbf{E}} + \mathbf{F}) = \mathbf{DF} + \mathbf{FF} + \mathbf{D}\overline{\mathbf{E}} + \overline{\mathbf{E}}\mathbf{F} =$$
$$\mathbf{DF} + \mathbf{F} + \mathbf{D}\overline{\mathbf{E}} + \overline{\mathbf{E}}\mathbf{F} = \mathbf{F}(\mathbf{D} + \mathbf{1} + \overline{\mathbf{E}}) + \mathbf{D}\overline{\mathbf{E}} = \mathbf{F} + \mathbf{D}\overline{\mathbf{E}}$$

1.9. Logikai függvények

A ***műszaki, technikai*** feladatok döntő hányada ***logikai döntések*** sorozatára épül. A logikai döntések elemei az állítások, amelyek értékei, és logikai kapcsolatuk határozza meg a döntések eredményét. A feladatokat megvalósító áramkörök, logikai hálózatok bemeneteire kapcsolt – az állításoknak megfelelő - kétértékű jelek a független logikai változók, míg a kimeneteken megjelenő – ugyancsak kétértékű – jelek a következtetések logikai értéke, és ezek a függő logikai változók. A ***függő-***, és a ***független*** változók közötti ***logikai kapcsolatot*** írják le a ***logikai függvények***. Minden függő változóra – kimeneti értékre – felírható egy-egy függvény.

A logikai függvény olyan ***egyenlőség***, amely ***változói kétértékűek***, és ezek között csak ***logikai műveleteket*** – ÉS, VAGY, TAGADÁS – végzünk.

A függvények megadása – leírása – történhet

- ***algebrai alakban,***
- ***táblázat segítségével,***
- ***matematikai jelölésekkel,***
- ***grafikus módon,***
- ***időfüggvény formájában.***

A felsorolt leírási módok teljesen egyenértékűek, és egymásba átírhatók!

A logikai kifejezések, függvények algebrai leírásának szabályait az 1.3. alfejezetben ismertettük. Az alábbiakban a további megadási formákat, és ezek kapcsolatát tárgyaljuk.

⇒ ***Logikai feladatok leírása táblázattal***

A logikai formában megfogalmazható, műszaki, számítási és irányítási feladatokban mindig *véges* számú ***elemi*** állítás szerepel. Ezek mindig ***csak két*** értéket vehetnek fel, vagy ***IGAZ*** - ak, vagy ***HAMIS*** - ak. Ebből következik, hogy a független változók lehetséges ***érték-variációinak*** a száma is *véges*. Minden egyes variációhoz a függő változó meghatározott értéke tartozik.

A logikai kapcsolat leírásának táblázatos formája az ***igazságtáblázat***. A táblázat tartalmazza a független változók összes kombináció-ját (érték-variációját) és az azokhoz rendelt függővváltozó(k) értékét, amit ***függvényértéknek*** is nevezhetünk. Az igazság-táblázatban minden logikai változó IGAZ értékét 1-el, míg a HAMIS értéket 0-val jelöljük.

Összefoglalva: ***az igazságtáblázat oszlopainak száma az összes logikai változó számával (függő változók száma + független változók száma), sorainak száma pedig a független változók lehetséges kombinációinak számával egyezik meg.***

A lehetséges értékvariációk számát (V-t) általánosan a $V=2^n$ összefüggéssel határoz-hatjuk meg, ahol n az összes független logikai változó száma.

Megjegyezzük, hogy általában csak egy függő változót tartalmazó igazságtáblázatot írunk fel. Azokban az esetekben, ha egy logikai kapcsolat-rendszerben több függő változó van, célszerűbb mindegyikre külön-külön felírni az igazságtáblázatot. Ezzel áttekinthetőbb képet kapunk.

A logikai alapműveletek igazságtáblázatait mutatja a 3. ábra.

K = A B

B	A	K
0	0	0
0	1	0
1	0	0
1	1	1

K = A + B

B	A	K
0	0	0
0	1	1
1	0	1
1	1	1

K = $\overline{A}$

A	K
0	1
1	0

3. ábra

6.Példa:

Írjuk fel a

$$\mathbf{Z = A\,\overline{B} + \overline{A}\,B}$$

logikai függvény igazságtáblázatát! A 3.ábrán követhető a leírt műveletsor.

Első lépésként az igazságtáblázat oszlopainak és sorainak a számát határozzuk meg. Mivel két független-, (A,B) és egy függő változó (Z) van, az oszlopok száma 3. (4.a.ábra). A sorok száma a független változók számából (n=2) a V = 2n = 22 = 4 összefüggésből számolható.

Második lépésként az értékvariációkat írjuk be. (4.b.ábra).Célszerű ezt úgy végrehajtani, hogy az egyik oszlopban (pl. az A) soronként váltjuk a 0, és az 1 beírását. A következő oszlopban (B) párosával váltogatjuk az értékeket. (Nagyobb sorszámnál a következő oszlopoknál négyesével, majd nyolcasával variálunk s.i.t.) A beírásnak ez a rendszeressége biztosítja, hogy egyetlen variáció sem marad ki.

B	A	Z

a.

B	A	Z
0	0	
0	1	
1	0	
1	1	

b.

B	A	Z
0	0	0
0	1	1
1	0	1
1	1	0

c.

4. ábra

Harmadik lépés az egyes sorokba írandó Z érték meghatározása. Ezt úgy végezhetjük el, hogy a független változóknak értékeket adunk, s az adott függvényt kiszámítjuk.

1. sorban: ***A = 0, B = 0***

Z = 0*1 + 1*0 = 0

2. sorban: ***A = 1, B = 0***

Z = 1*1 + 0*0 = 1

3. sorban: ***A = 0, B = 1***

Z = 0*0 + 1*1 = 1

4.sorban: ***A = 1, B = 1***

Z = 1*0 + 0*1 = 0

A példa szerinti logikai függvény igazságtáblázata a 4.c.ábrán látható.

Az előző példa egy sokszor használt függvény-kapcsolat, az un. ***KIZÁRÓ-VAGY*** (XOR) művelet. (Nevezik moduló összegnek is.) A művelet eredménye akkor 1, ha a két változó közül az egyik 1. Több változóval is végezhető ***moduló - összegzés***, és eredménye akkor 1, ha ***páratlan számú*** független változó értéke 1.

⇒ *Logikai függvény felírása az igazságtáblázatból*

Az előző pontban megismerkedtünk az igazság-táblázattal, amely a logikai kapcsolatrendszer leírásának egyik formája. Példa segítségével mutattuk be, hogy ismert logikai függvényből hogyan írható fel a táblázatos alak.

Ebben a részben azt tárgyaljuk, hogy ha ismert az igazságtáblázat, hogyan lehet abból felírni a logikai függvényt.

Az igazságtáblázat egy sora a független változók adott kombinációját, és az ehhez tartozó függvény értékét adja.

Az egy sorban levő értékeket az ÉS művelettel lehet összekapcsolni. A különböző sorok pedig különböző esetnek megfelelő variációkat írnak le. Tehát egy adott időpillanatban vagy az egyik sor vagy egy másik sor variációja érvényes. A sorok logikai kapcsolata VAGY művelettel írható le. Vegyük példaként az 5.ábrán látható igazságtáblázatot.

C	B	A	K
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

5. ábra

A táblázatból kétféle alakú függvény írható fel a következő állítás alapján:

A függvényérték **IGAZ**

- ***azokban*** a sorokban, amelyekben a függő változó ***1***, illetve
- ***nem*** azokban a sorokban ahol függő változó ***0***.

Az állítás első fele szerint fel kell írni az 1 értékhez tartozó sorok változókombinációinak VAGY kapcsolatát.

A második rész szerint a 0 értékhez tartozó sorokhoz tartozó változókombinációinak VAGY kapcsolatát, majd az egyenlőség mindkét oldalát tagadni kell.

A példa szerinti igazságtáblázatból írjuk fel először a független változók 1 értékeihez tartozó függvény algebrai alakját.

Az igazságtáblázat tartalmát a következőképpen olvassuk ki. A K jelű függő változó értéke 1 (IGAZ),

ha C = 0 és B = 0 és A = 1 (2.sor),vagy

ha C = 0 és B = l és A = 0 (3.sor),vagy

ha C = 0 és B = 1 és A = 1 (4.sor), vagy

ha C = l és B = l és A = 0 (7.sor).

Az A,B,C és K változók közötti logikai kapcsolat az előbbiek szerint

$$\mathbf{K = A\overline{B}\overline{C} + \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC}$$

alakban írható fel.

A függvény rendezett ÉS-VAGY alakú. Az ÉS művelettel összekapcsolt részekben mindegyik változó szerepel egyenes (ponált) vagy tagadott (negált) alakban, vagyis a Veitch diagramnál definiált minterm.

Az egyes minterm -ek között pedig VAGY műveleteket kell végezni. Az ilyen függvényalakot idegen szóval diszjunktív kanonikus alaknak (teljes diszjunktív normál formának) nevezzük.

A felírás szabálya a következő:

1. azokat a sorokat kell figyelembe venni, amelyeknél a függő változó értéke 1;

2. az egy sorban levő független változók között ÉS műveletet kell végezni, ahol a független változó igaz (egyenes, más kifejezéssel ponált) alakban írandó, ha értéke 1 és tagadott (negált) alakban, ha értéke 0;

3. az egyes sorokat leíró ÉS műveletű rész-függvények VAGY művelettel kapcsolódnak egymáshoz.

A kiinduló állítás második része szerint:

Azt nézzük meg, hogy mikor nem IGAZ (HAMIS) a következtetés.

A K a következő kombinációknál (sorokban) 0, (vagyis $\overline{K}$)

ha C=0 és B=0 és A=0 (1.sor) vagy

ha C=1 és B=0 és A=0 (5.sor) vagy

ha C=1 és B=0 és A=1 (6.sor) vagy

ha C=1 és B=1 és A=1 (8.sor).

A leírt logikai kapcsolatot a

$$\overline{K} = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}\overline{C} + ABC$$

függvénnyel írhatjuk le. Ebből a K értékét mindkét oldal tagadásával nyerhetjük.

$$\overline{\overline{K}} = \overline{\overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}\overline{C} + ABC}$$

A baloldalon K-t kapunk. A jobb oldal átalakítását a de Morgan - tételek alkalmazásával végezhetjük el.

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

A kapott függvényt elemezve, megállapíthatjuk, hogy a függvény VAGY-ÉS alakú. A zárójeles VAGY műveletek mindhárom független változót (A,B,C) tartalmazzák egyenes vagy tagadott alakban. Ezek maxterm -ek, melyeket a Veitch diagramnál definiáltunk. Az első maxterm az igazságtáblázat első sora szerinti állítás - vagyis, hogy

az A=0 és B=0 és C=0 - tagadása. A további tagokat vizsgálva látjuk, hogy ezek is egy-egy olyan sornak a tagadásai, melyben K=0.

Az előzőek alapján most már megfogalmazhatjuk, hogy az igazságtáblázatból úgy is felírhatjuk a feladatot leíró logikai függvényt, hogy

1. azokat a sorokat vesszük figyelembe, melyekben a függő változó értéke 0;

2. az egy sorban levő független változók között VAGY kapcsolatot írunk elő;

3. a független változót egyenes alakban írjuk, ha értéke 0 és tagadott alakban, ha értéke 1;

4. az egyes sorokat leíró VAGY függvényeket ÉS művelettel kell összekapcsolni.

Azt a logikai függvényt, amely maxtermek logikai szorzata idegen szóval konjunktív kanonikus alakúnak, rendezett VAGY-ÉS függvénynek (teljes konjunktív normál alakúnak) nevezzük.

⇒ ***Logikai függvények matematikai, egyszerűsített felírási alakjai***

Mivel a logikai változónak két értéke – 0, illetve 1 – lehet, ezért ezt tekinthetjük egy bináris számjegy -nek is.

A függvény egy - egy maxterm – jét, vagy minterm - jét, oly módon is leírhatjuk, hogy az hányadik eleme a mintermek, illetve maxtermek rendezett sorának.

A sorszám kiszámolásához első lépésként a változókhoz a bináris számrendszer egy-egy helyértékét kell hozzárendelnünk, vagyis súlyozunk. Azért tehetjük ezt, mert a logikai változók értéke 0, vagy 1 lehet, és a változók kombinációinak értéke formailag egy bináris számot alkotnak.

A súlyozás kiválasztása után az egyes kombinációkban a ponált változó helyére 1-t, míg a negált helyére 0-t írunk. Az így kapott szám lesz az adott maxterm, vagy minterm sorszám-a (súlya). A számolást bináris számrendszerben végzzük, de az indexet decimálisan fogjuk írni, mivel ez kevesebb helyet igényel.

7.Példa:

Legyen a $C \div 2^2, B \div 2^1, A \div 2^0$ súlyozású. Ekkor a

$C\overline{B}A$ minterm súlya: $1*2^2 + 0*2^1 + 1*2^0 = 101_B = 5$

a $\overline{C} + B + \overline{A}$ maxterm súlya: $0*2^2 + 1*2^1 + 0*2^0 = 010_B = 2$.

A mintermeket az m_i^v jelöléssel helyettesíthetjük, ahol az m jelzi, hogy a logikai egység minterm, a felső index v a változók számát, az alsó index i pedig a sorszámot jelenti.

Hasonlóan a maxterm -eket is helyettesíthetjük a M_i^v jelöléssel. Az indexek (v,i) jelentése ugyan az, míg az M jelzi, hogy a logikai kifejezés maxterm.

A leírtakat a példában szereplő kifejezésekre (ugyanazon változó súlyozásnál) a

$$C\overline{B}A \div m_5^3$$

és a

$$\overline{C} + B + \overline{A} \div M_2^3$$

helyettesítéseket alkalmazhatjuk.

⇒ ***Függvények megadása matematikai alakban***

Az ismertetett helyettesítésekkel a diszjunktív, valamint konjunktív kanonikus alakú függvények is rövidebben leírhatóak. Vegyük példának az előzőekben felírt függvények alaki helyettesítését az $A \div 2^2, B \div 2^1, C \div 2^0$ változó súlyozás alkalmazásával:

$$K = A\overline{B}\overline{C} + \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC$$

$$K = m_4^3 + m_2^3 + m_6^3 + m_3^3$$

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

$$K = M_7^3 * M_6^3 * M_2^3 * M_0^3$$

A függvények felírása tovább is egyszerűsíthető oly módon, hogy

megadjuk a függvény – alak -ot

a változók számát, és

a függvényben szereplő term –ek sorszámait.

A diszjunktív alakot a $\sum^{v}(.....)$,a

konjunktív alakot a $\prod^{v}(.....)$ formában írjuk.

A két minta függvény egyszerűsített felírása (ugyanazon változó-súlyozást alkalmazva):

$$\mathbf{K = \sum^{3}(2,3,4,6)}$$

$$\mathbf{K = \prod^{3}(7,6,2,0)}$$

⇒ ***Kanonikus függvény-alakok közötti átalakítás***

Az előzőekben megismertük, hogyan lehet a logikai feladat igazságtáblázatából felírni a logikai függvény két kanonikus alakját.

Az egyik kanonikus alakú függvény egyszerűsített (indexelt) formája alapján nagyon egyszerűen felírható a másik rendezett alak egyszerűsített formája.

Az átalakítás menete a következő:

az ismert függvény alapján felírjuk az inverz függvényt (amely az alap függvény tagadottja), ezt a hiányzó indexű term – ek alkotják,

pl. ha ismert a diszjunktív alak:

$$\mathbf{K = \sum^{3}(2,3,4,6)} \qquad \Rightarrow \qquad \mathbf{\overline{K} = \sum^{3}(0,1,5,7)}$$

ismert a konjunktív alak:

$$\mathbf{K = \prod^{3}(7,6,2,0)} \qquad \Rightarrow \qquad \mathbf{\overline{K} = \prod^{3}(5,4,3,1)}$$

az inverz függvény tagadásával nyerjük a másik alakú rendezett függvényt. A tagadáskor a függvény - típusjele az ellenkezője lesz, és mindegyik index (i) B-1 –es kiegészítőjét ($\bar{\mathbf{i}}$) kell vennünk a következő összefüggés alapján:

$$\bar{\mathbf{i}} = (2^{v} - 1) - \mathbf{i}$$

A tagadások elvégzése után

$$\mathbf{\overline{\overline{K}} = \overline{\sum^{3}(0,1,5,7)}} \qquad \Rightarrow \qquad \mathbf{K = \prod^{3}(7,6,2,0)}$$

$$\mathbf{\overline{\overline{K}} = \overline{\prod^{3}(5,4,3,1)}} \qquad \Rightarrow \qquad \mathbf{K = \sum^{3}(2,3,4,6)}$$

megkaptuk a keresett alakú függvényeket.

⇒ *A logikai függvények grafikus megadása*

A logikai függvények gyakori ábrázolási módjai:

a logikai műveletek szimbólumaival megrajzolt logikai vázlat,

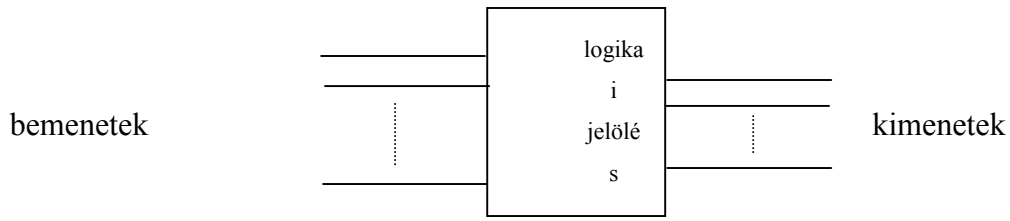
síkban, vagy térben a Veitch diagramból származtatott minterm -, és maxterm – diagram, illetve a Karnaugh – diagramok segítségével,

az idő függvényében rajzolt grafikon formájában.

⇒ *Logikai vázlat*

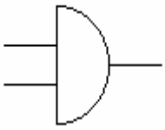
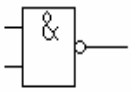
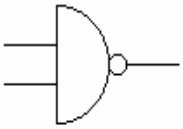
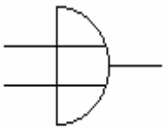
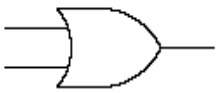
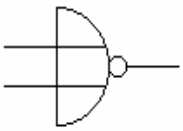
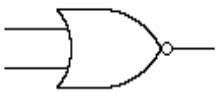
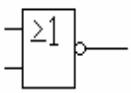
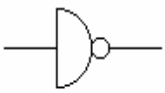
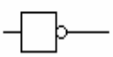
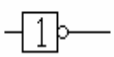
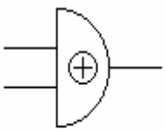
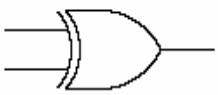
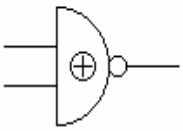
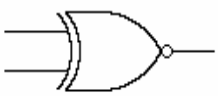
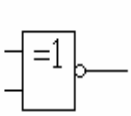
A szimbólumokkal történő ábrázolás az áramköri megvalósítást segítő megoldás, amelyet az elmúlt fél évszázadban több változatban is szabványosítottak. Az érvényes európai, és hazai szabványok közös jellemzői:

- ***a szimbólum kerete négyszög,***
- ***a négyszögbe írt jelölés utal a logikai funkcióra,***
- ***a független változókat jelző bemenetek a keret bal oldalához,***
- ***míg a függő változókat jelző kimenetek a keret jobb oldalához csatlakoznak.***



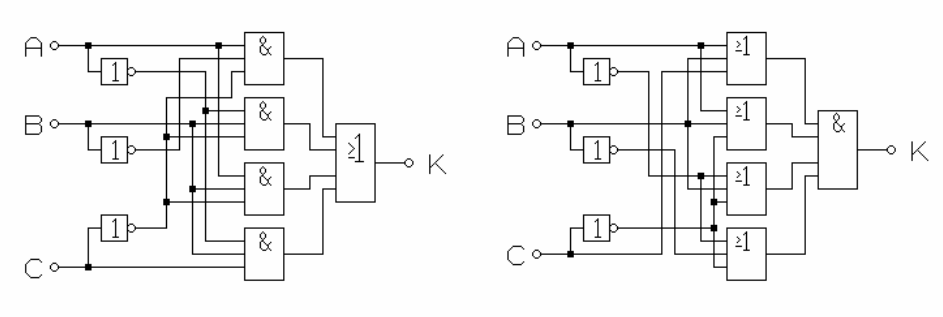
A be-, és kimenetek jeleit általában a csatlakozó vezetékre kell írni. (Ettől eltérő felírással az összetett szimbólumoknál találkozunk.)

Nemzetközileg a szabványosítást az 1970 – es években kezdték el. Addig országonként, gyártó cégenként szabványosított szimbólumokat használtak. A módokról, és azok változásáról a mellékletben adunk áttekintést. A 6.ábrán csak a logikai alapműveleteket szemléltető szimbólumokat mutatjuk be.

	Magyarországon 1950-60	TEXAS jelölések 1967-től	1975-től szabványos
ÉS (AND)			
ÉS-NEM (NAND)			
VAGY (OR)			
VAGY-NEM (NOR)			
NEM (INVERS)			
KIZÁRÓ-VAGY (XOR)			
KIZÁRÓ-VAGY-NEM (NXOR) máskép EGYENLŐ (EQUALENCIA)			

6. ábra

A fejezetben példaként felírt függvény kétféle kanonikus alakjának logikai vázlatát mutatja a 7.a. és b. ábrák.



a. b.

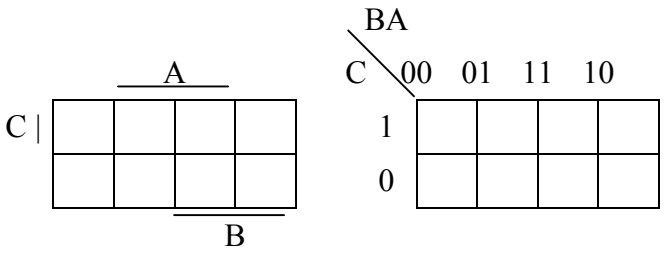
7. ábra

1.10. Grafikus ábrázolás

⇒ ***Karnaugh diagram***

A grafikus ábrázolásainak egyik változata, hogy logikai sík-, vagy térbeli ***geometriai alakzatot*** rendelünk.

A függvényhez rendelt geometriai alakzat ***peremén*** adjuk meg a logikai ***változók jeleit***. Ezzel adjuk meg azt, hogy az alakzat melyik részén ***IGAZ*** értékű ez a változó. (Az alakzat másik részén – értelemszerűen – a változó HAMIS értékű.). Ezt a jelölés-rendszert ***peremezésnek*** nevezzük. A ***binárisan*** kódolt peremezésű változatot nevezzük ***Karnaugh*** táblázatnak. Használják még az ***oldal mellé húzott*** vonallal történő peremezést is. A tanulmányainkban a Karnaugh táblázatot fogjuk használni, mivel az igazságtáblázatból történő átírás egyszerűbb. A 8.ábra három változós (A B C) logikai függvény megadásához használható síkbeli elrendezés kétféle peremezését mutatja.



8. ábra

Mindkét változat formailag a Veitch diagramból származtatott. A különbségek a változók megadásának (a peremezésnek) módjában, valamint abban van, hogy egy elemi négyszög ***mintermet***, vagy ***maxtermet*** is jelképezhet. Egy n változós függvény 2^n db elemi négyzetből álló táblázatban szemléltethető.

Az eljárás a 8.ábra alapján követhető. A halmazt egy négyszögben ábrázoljuk. Minden változó IGAZ értékéhez a teljes terület egyik felét, míg a HAMIS értékéhez pedig a másik felét rendeljük. Az értékeket a négyszög szélére irt, ***vonallal*** (minterm / maxterm tábla vagy diagram), illetve ***kódolással*** (Karnaugh-diagram) adjuk meg. A továbbiakban a Karnaugh - diagramot használjuk. Több változó esetén a felezést úgy folytatjuk, hogy a változókhoz rendelt területeket jól meg lehessen különböztetni. A változók kódolását (kijelölését) úgy kell végezni, hogy az ***egymás melletti oszlopok***, ill. ***sorok*** mindig ***csak egy*** változóban ***térjenek*** el egymástól. A Hamming - távolság 1.

A háromváltozós Karnaugh - táblázat oszlopaihoz a BA változó-pár lehetséges érték-kombinációt rendeltük. Az oszlop-peremezést úgy kell végezni, hogy a szomszédos oszlopok csak egyetlen változó-értékben különbözzenek. A harmadik változó C értéke szerint két sora van a táblázatnak. Az egyikben C=0, a másikban pedig C=1. Az egyes elemi négyszögekhez tehát a változók különböző értékvariáció tartoznak. A peremezés megváltoztatható, de csak úgy, hogy a szomszédos sorok, oszlopok egy változóban különbözhetnek. (A táblázat szélső oszlopai, illetve sorai mindig szomszédosak).

A 9.ábrán a négy változós Karnaugh diagram látható

DC \ BA	00	01	11	10
00				
01				
11				
10				

9. ábra

A 10.ábrán az 5, a 11.ábrán pedig a 6 változós táblázatot láthatjuk. (Az ábrázolási mód legfeljebb 6 változóig alkalmazható szemléletesen.)

Az öt-változós táblázatot célszerű két négy-változós táblázatból úgy kialakítani, hogy a két rész peremezése csak az egyik változóban - itt pl. a C – tér el egymástól.

CBA / ED	000	001	011	010	100	101	111	110
00								
01								
11								
10								

10. ábra

A 6 változós táblázatnál függőlegesen duplázzuk meg a táblázat elemeit.

CBA / FED	000	001	011	010	100	101	111	110
000								
001								
011								
010								
100								
101								
111								
110								

11. ábra

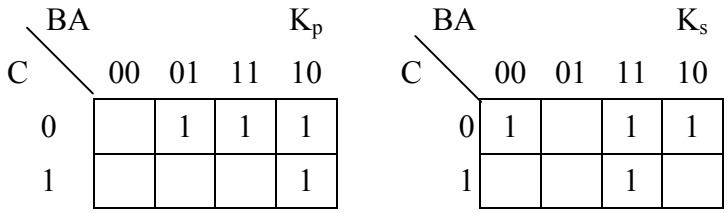
Így négy egyforma 4 változós egységeket kapunk. Az egyes rész-táblázatokban négy változót (ABED) azonosan variálunk. Az eltérés vízszintesen a C, míg függőlegesen az F változó.

Az eddigiekben csak az ábrázolás formai részével foglalkoztunk. Nézzük most meg a logikai tartalmat is. A két hozzárendelés szerint beszélünk ***Kp*** ill. ***Ks*** diagramról. A ***p*** index arra utal, hogy az elemi cellában logikai ***szorzat*** (produktum), míg az ***s*** a logikai ***összeget*** jelenti (summa). Tehát a ***Kp*** jelölés az ***ÉS-VAGY***, míg a ***Ks*** a ***VAGY-ÉS*** műveletes teljes függvényalakot adja meg.

A logikai függvényt ***diszjunkt*** alakját úgy kell a ***Kp*** diagramban ábrázolni, hogy a függvényben szereplő ***mintermeket*** reprezentáló cellákba ***1***-et írunk.

A ***konjunkt*** alakot ***Ks*** diagramban ábrázoljuk oly módon, hogy a megfelelő ***max-termeket*** jelentő cellákba írunk ***1***-t. (A 0-t egyik változatban sem szokták kiírni, a cella üres).

A fejezetben már leírt példa Karnaugh diagramjai láthatók a 12.a. és b. ábrákon.



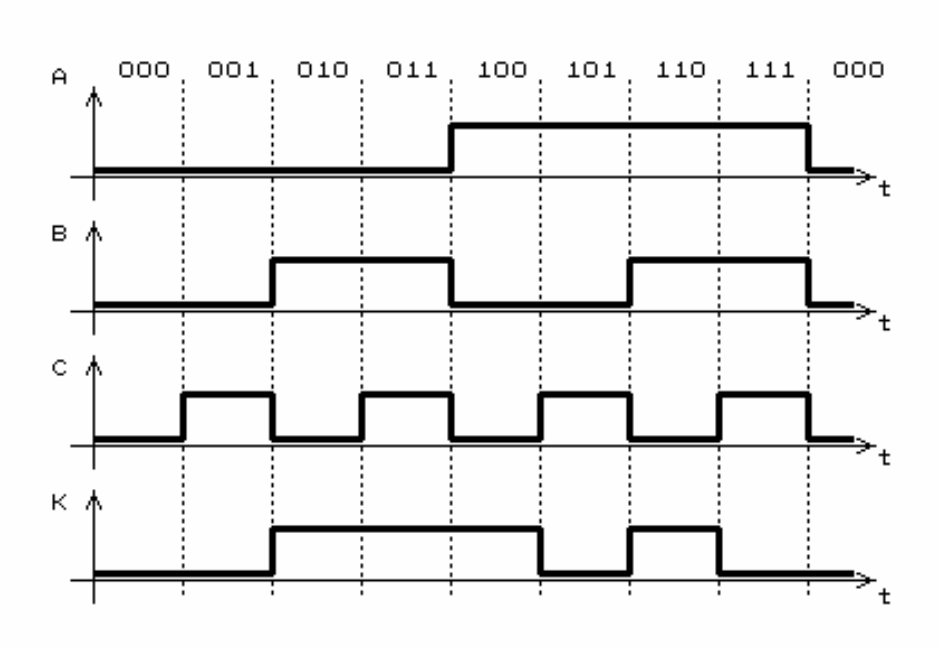
12. ábra

⇒ ***Időfüggvény megrajzolása***

A függvény minden változójának ***időbeli lefolyását*** ábrázoljuk ***fázishelyesen*** egy-egy derékszögű ***koordináta*** rendszerben. A módszert elsődlegesen az egyes digitális áramkörök vizsgálatánál alkalmazzuk oly módon, hogy a bemeneteket (független változókat) ismert digitális jelekkel gerjesztjük. Az áramkör kimenetén – oszcilloszkóppal - mért jel a függvény értékének változását adja meg. A be-, és kimenetek jeleiből a vizsgált áramkör logikai függvényének bármelyik alakja meghatározható.

A fejezetben már ismert logikai függvény be-, és kimeneteinek időfüggvényét mutatja a 13. ábra. A bemeneteket bináris kód szerint változó kombinációsorozattal gerjesztjük

A szaggatott vonalak jelzik a gerjesztések változásának időpontjait. A matematikai leírásnál használt változó-súlyozással irtuk fel az egye kombináció bináris sorszámát. Ebből közvetlenül kiovasható, hogy a K kimenet IGAZ értékű lesz, ha a bemeneteket a 2, 3, 4, és 6 sorszámú kombinációk valamelyike gerjeszti.



13. ábra

1.11. A logikai függvények egyszerűsítése

Az igazságtáblázat alapján felírt kanonikus alakú függvények a legtöbb esetben ***redundánsak***, tehát egyszerűsíthetőek. A redundancia azt jelenti, hogy a megadott információ több, mint amennyi az egyértelmű függvényleíráshoz szükséges.

Az ***egyszerűsítés*** során a logikai algebra megismert tételeinek felhasználásával olyan alakot nyerhetünk, amelyben ***kevesebb művelet***, és vagy kevesebb ***változó*** szerepel. Az egyszerűsítésre azért van szükség, mert ez után a feladatot megvalósító logikai hálózat kevesebb áramkört, vagy programozott rendszer (mikrogép) programja kevesebb utasítást tartalmaz

Az ***algebrai*** módszer mellett kidolgoztak ***grafikus***, illetve ***matematikai*** egyszerűsítési eljárásokat is.

A felsorolt egyszerűsítési (minimalizálási) eljárásokat a fejezetben bemutatott igazságtáblázattal leírt logikai feladat segítségével ismertetjük.

A 14. ábrán látható feladat igazságtáblázata:

C	B	A	K
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

14. ábra

⇒ ***Algebrai egyszerűsítés***

A logikai algebra tárgyalásakor már bemutattunk néhány átalakítási eljárást. Itt egy újabb példa segítségével végezzük el a feladat legegyszerűbb alakjának megkeresését.

a. Egyszerűsítés a ***diszjunktív*** alakú függvényből

$$\mathbf{K = A\overline{B}\overline{C} + \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC}$$

Először keressük meg, hogy vannak-e közös részeket tartalmazó ***mintermek***. Ezekből ”emeljük” ki a közös részeket!

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

A zárójelekben lévő mennyiségek értéke ***1***, ezért azok a logikai szorzatból elhagyhatók. A keresett, legegyszerűbb függvényalak a következő:

$$\mathbf{K = \overline{A}B + A\overline{C}}$$

b. Egyszerűsítés ***konjunktív*** alakú rendezett függvényből

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

Hasonlóan az előző egyszerűsítéshez itt is végezhetünk – a disztributív tulajdonság alapján - ”kiemeléseket” a ***maxterm*** - ekből.

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

A $C\overline{C}$ és $B\overline{B}$ tényezők értéke ***0*** és ezért a logikai összegekből elhagyhatók. A keresett legegyszerűbb függvényalak tehát:

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

c. ***Igazoljuk*** a két alakból kapott függvények ***azonosságát***, vagyis hogy igaz az

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

egyenlőség.

Végezzük el a jobb oldalon a "***beszorzást***"!

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

A kapott kifejezésben az első tényező 0. A negyedik tényezőt "***szorozzuk***" 1-el.

$$0 + \overline{A}B + A\overline{C} + B\overline{C}(A + \overline{A}) = \overline{A}B + A\overline{C} + B\overline{C}A + B\overline{C}\overline{A}$$

A közös részek "***kiemelése***" után

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

a zárójeles kifejezések elhagyhatók, mivel értékük 1. A kapott eredménnyel igazoltuk az eredeti egyenlőség azonosságát.

Ezzel bizonyítottuk, hogy az igazságtáblázatból a két - ismertetett - módszer bármelyikével ugyanazt a függvényt kapjuk.

Összefoglalva: ***megállapíthatjuk, hogy az igazságtáblázatból rendezett ÉS-VAGY (diszjunktiv kanonikus) alakú vagy rendezett VAGY-ÉS (konjunktív kanonikus) alakú logikai függvényt írhatunk fel. A két alak azonos függvényt ír le.***

⇒ *Grafikus egyszerűsítés Karnaugh –táblázattal*

A leírt kikötések betartásával - az előző fejezetben megismert - mindkét logikai függvényalak (diszjunktiv, ill. konjunktív) ábrázolható, és egyszerűsíthető Karnaugh – diagram segítségével. A Karnaugh diagramok – min ahogyan azt az előző fejezetben megismertük - az igazságtáblázatból közvetlenül felírhatók.

a. Kp diagram használata.

A Karnaugh diagram egyes celláiba kell beírni a független változók (A,B,C) megfelelő kombinációihoz tartozó függő változó (K) értéket (15.ábra).

Az A=0,B=0,C=0 kombinációnál a K értéke 0, tehát a BA=00 oszlop és C=0 sor által meghatározott cellába 0-t kell írni és így tovább.

C \ BA	00	01	11	10 (K_p)
0		1	1	1
1				1

15. ábra

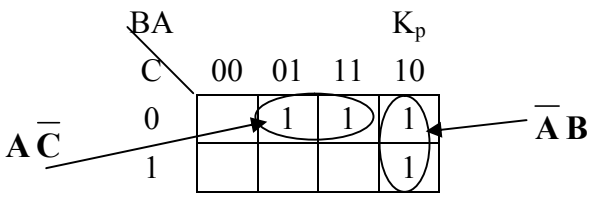
A 0 értékeket nem fontos beírni, ugyanis az egyszerűsítésnél csak az 1 értékű cellákat vesszük figyelembe.

Vizsgáljuk meg a diagram ***utolsó oszlopában*** lévő ***két*** cella tartalmát.

A ***felső*** cella tartalma az $\overline{A}\overline{B}\overline{C}$, míg az ***alsó*** celláé $\overline{A}BC$ ***minterm***. Mivel mindkét cella értéke 1, azt jelenti, hogy mindkét minterm a függvény tagja, és közöttük VAGY kapcsolat van. A két minterm -ből álló függvényrész egyszerűsíthető.

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

A példa alapján is bizonyítottnak tekinthetjük, hogy ha ***két*** – ***élben érintkező*** – cellában 1 van, akkor ezek ***összevonhatók***, vagyis ***az a változó kiesik,*** amelyikben ***különböznek*** a cellák. Az összevonhatóságot ***lefedő hurokkal*** szokás jelölni (16.ábra):



16. ábra

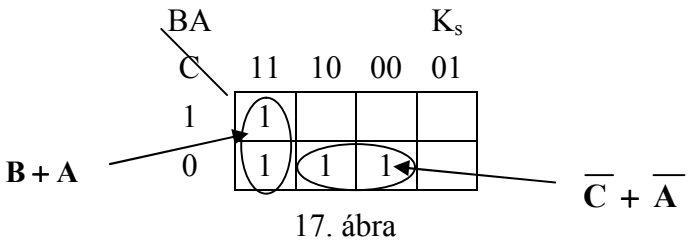
A lefedett (összevont) cellák VAGY kapcsolata adja az egyszerűsített függvényt:

$$\mathbf{K = \overline{A}B + A\overline{C}}$$

b. Ks diagram használata.

Az egyszerűsített függvényalakoknál tárgyaltakhoz hasonlóan a ***Kp*** és a ***Ks*** diagramok is ***felrajzolhatók egymásból***.

Az átrajzolásnál a ***peremezés***, és a cella-értékek ***komplemens*** -ét kell írni, vagyis 0 helyett 1-e, és fordítva. A 17.ábrán látható a példa Ks diagramja:



17. ábra

A cellák most maxtermeket tartalmaznak, ezért az egyszerűsített függvény az összevonások (lefedések) közötti ÉS művelettel írható le:

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

c. Több cella összevonása.

A logikai függvények között vannak olyanok is, melyeknél ***többszörös*** algebrai ***összevonás*** is végezhető.

Keressük meg a következő négy (A,B,C,D) változós logikai függvény legegyszerűbb alakját!

A változókat súlyozzuk az $\mathbf{A \div 2^0, B \div 2^1, C \div 2^2, D \div 2^3,}$ szerint. A függvény egyszerűsített alakja:

$$\mathbf{F = \sum^{4}(8,10,12,13,14,15)}$$

Rajzoljuk meg a függvény Karnaugh táblázatát (18.ábra).

DC \ BA	(0) 00	(1) 01	(3) 11	(2) 10
(0) 00				
(4) 01				
(12) 11	1	1	1	1
(8) 10	1			1

18. ábra

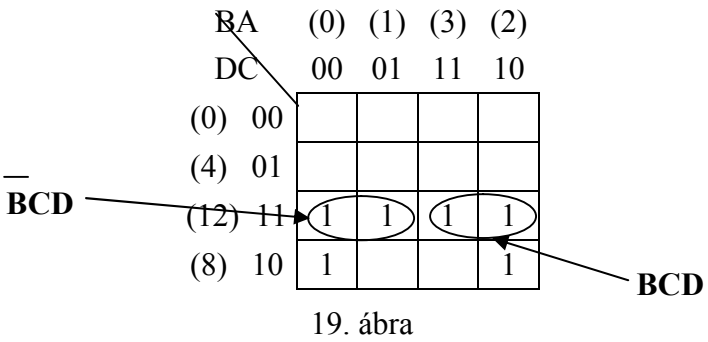
A Karnaugh diagram – egyszerűsített alakú függvény alapján történő – felrajzolását könnyíti, ha az egyes ***sorok*** és ***oszlopok súlyát*** decimálisan is jelöljük. Ezt tettük a zárójelbe írt számokkal.

Először írjuk fel a harmadik sor rész-függvényét algebrai alakban, mivel mindegyik cellában 1 értékű a függvény.

$$\overline{A}\overline{B}CD + \overline{A}BCD + ABCD + A\overline{B}CD = \overline{B}CD(\overline{A} + A) + BCD(A + \overline{A}) =$$
$$= \overline{B}CD + BCD = CD(\overline{B} + B) = CD$$

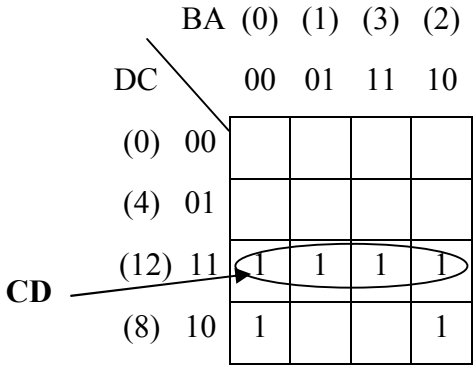
Az algebrai sorozatos kiemelések után két változó (A,B) kiesett.

Ugyanezt kövessük végig Karnaugh diagramon is. A 19.ábrán az első egyenlőségjel utáni két kettős összevonás látható.



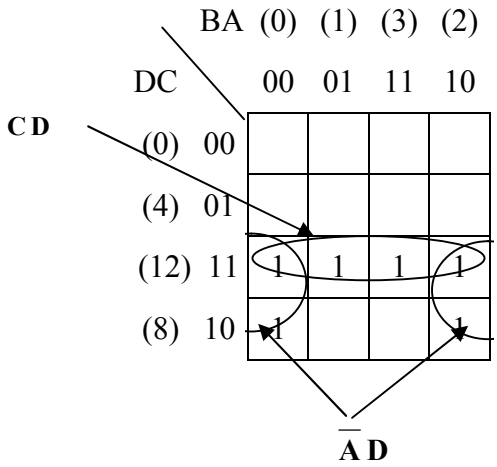
19. ábra

Mindkét lefedésnél kiesett az A változó. A két háromváltozós rész-függvényben közös a CD logikai szorzat, tehát összevonható. A grafikus módszernél ez egy közös lefedéssel jelölhető (20.ábra).



20. ábra

Hasonló négyes csoportot alkotnak a 8,10,12,14 sorszámú mintermek is, tehát összevonhatók (21.ábra).

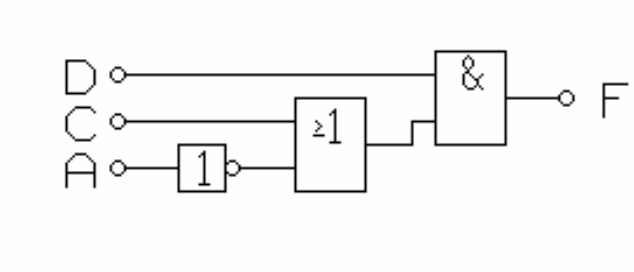


21. ábra

Az egyszerűsített függvény a két részfüggvény logikai összege, amely még algebrailag tovább egyszerűsíthető:

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

Az utolsó egyszerűsítés eredményeként kaptuk a legkevesebb művelettel megvalósítható alakot. A függvény logikai vázlata látható a 22.ábrán.



22. ábra

Összefoglalás:

A grafikus függvényegyszerűsítés szabályi:

- ***a lefedhető (összevonható) cellák száma 2^n (n pozitív egész szám), ha azok kölcsönösen szomszédosak,***
- ***a kölcsönösen szomszédos meghatározást úgy kell érteni, hogy a kiinduló cellától kezdve a élben érintkező szomszédos cellákon keresztül 2^n számú lépés után az kiindulóhoz jutunk vissza,***
- ***a lefedett cellákból a kitevőnek (n) megfelelő számú változó esik ki, amelyek a lefedés alatt változnak,***
- ***minden 1 -t tartalmazó cellát legalább egyszer le kell f e d n i .***

1.12. Aritmetikai alapfogalmak

A digitális berendezésekben – mérőegységek, számítóművek stb. – gyakori feladat ***aritmetikai műveletek*** végzése. Az eddig megismert logikai műveletek változói kétértékűek. A számok ***bináris*** – kettes – ***számrendszerben*** való ábrázolásánál is a ***0***, és az ***1*** számjegyeket használjuk. A későbbiekben igazoljuk, hogy az aritmetikai műveletek elvégzése logikai műveletekkel lehetséges. Itt most összefoglaljuk – az alábbi - alapvető ***aritmetikai fogalmakat:***

- ***szám, számjegy, számrendszer,***
- ***számábrázolási formák,***
- ***aritmetikai alapműveletek algoritmusai.***

⇒ *Szám, számjegy, számrendszer*

Röviden összefoglaljuk – a korábbi tanulásaikban már megismert – fogalmakat.

- A ***szám***:

A ***szám*** „valaminek” a számosságát, mennyiségét, értékét megadó ***jelcsoport***. A jelcsoportok mind a használt ***jelek***, mind a ***jelölési-rendszer*** felépítése szerint változtak az idő folyamán.

- A ***számjegy***:

A ***számjegyek*** a számként használt jelcsoport egyes ***jelei***, amelyekhez ***konkrét értéket*** rendeltek. Egy jelölési-rendszeren belül véges számú számjegy van.

- A ***számrendszer***:

A ***számrendszer*** határozza meg, hogy a használt ***jelekből*** milyen ***módon***, (algoritmus szerint) kell ***leírni*** (ábrázolni) egy ***számot***. A számrendszerek a korai időszakokban kultúránként különböztek. A tudományok, a technika fejlődésének eredményeként egységes számrendszerekről beszélhetünk.

- A ***RÓMAI*** számrendszer

A mai napig szélesebb körben is ismert számrendszert a rómaiak alkották meg. A ***római*** számokban a következő *7* ***számjegy*** (jel) létezik (A zárójelbe írjuk a jelhez rendelt értéket tízes számrendszer szerinti jelöléssel.)

Számjegyek és értékük

I (1)	**X** (10)	**C** (100)	**M** (1000)
V (5)	**L** (50)	**D** (500)	

A számjegyek megfelelő szabályok szerinti egymás utáni írásával fejezték ki a szám-értékeket. Tulajdonképpen a tízes váltószám, amely valószínűleg ujjaink számából ered megtalálható a számrendszer logikájában. A számalkotás szabályát itt nem részletezzük, csak egy példával illusztráljuk.

$$\mathbf{M\ C\ M\ L\ XX\ IV = 1974}_{10}$$

A romai számok segítségével értékeket - korlátozott terjedelemben – ki lehet fejezni. Számtani műveletek ezekkel nem végezhetők.

- A ***strukturált*** számrendszerek

Pontosan nem ismert, hogy a mai értelemben vett számrendszerek alapjait mikor és hol fektették le. Az európai kultúrában, és az abból építkezőkben használt számjegyek arab eredetűek.

A ma használt számrendszerek egy-egy ***alapszámra*** épülnek, és a számjegyek ***száma*** az alapszám ***értéke***. Felépítésük, pedig az alapszám egész számú hatványa - ***helyérték*** – szerint tagolódik. Általános leírása:

$$Z = (x_{n-1} B^{n-1} + x_{n-1} B^{n-1} + ... + x_1 B^1 + x_0 B^0) + (x_{-1} B^{-1} + x_{-2} B^{-2} + \ldots + x_{-p} B^{-p})$$

egész rész tört rész

ahol,

B	a számrendszer ***alapszáma,***
x_i	az i - ik helyérték ***számjegye*** ($0 \leq x \leq B-1$),
n	az ***egész*** rész helyértékeinek száma,
p	a ***tört*** rész helyértékeinek a száma.

- A ***leggyakrabban használt*** számrendszerek:

	alapszám	***számjegyek***
Tízes (decimális)	B = 10	0, 1, …8, 9
Kettes (bináris)	B = 2	0, 1
Nyolcas (oktális)	B = 8	0, 1,…6, 7
Tizenhatos (hexadecimális)	B = 16	0, 1, …. 9, A, B, C, D, E, F

Ismert módon a számok felírásánál csak az egyes helyértékekhez tartozó számjegyeket írjuk balról-jobbra, a legnagyobb helyértékű számjeggyel kezdve. A különböző alapszámok, valamint a részben azonos számjegyek miatt, a számoknál jelezni kell, hogy az milyen számrendszerben értendő. A jelzést lehet a szám ***előtt*** – ***prefix*** -, vagy a

szám ***után*** – ***suffix*** – megadni. Legtöbb esetben a decimális számokat jelzés nélkül írják.

pl.

számrendszer	jel nélkül	prefix	suffix	
decimális:	1456	**0d** 1456	1456 **d**	1456_{10}
bináris	-	**0b** 100110	100110 **b**	100110_{2}
oktális	-	**0o** 273	273 **q**	273_{8}
hexadecimális	-	**0x** 1A2D	1A2D **h**	$1A2D_{16}$

Megjegyzés: a jelzőkben, illetve számjegyekként használt betűk kis-, és nagybetűk is lehetnek. A ***hexadecimális*** számoknál, ha azok betűvel kezdődnek, akkor egy ***0***-t kell írni a ***szám elé,*** pl. 0A4CF.

- A számok ***komplemens*** -e (***kiegészítő*** -je).

A szám ***komplemens*** -e (kiegészítője) - mint a neve is utal rá - az érték, amely a számot ***kiegészíti*** a számrendszer egy ***adott értékéhez***. A definíció szerint bármely értékhez számolhatnánk a kiegészítőt, de gyakorlati jelentősége csak az alábbi két változatnak van.

A kiegészítés történhet:

- a szám ***nagyságrendjébe*** tartozó ***legnagyobb*** értékéhez, vagyis a (B^n-1) - hez,
- a számnál ***egy nagyságrenddel*** nagyobb ***legkisebb értékéhez***, vagyis a B^n - hez,

ahol **B** az alapszám, és **n** a nagyságrendek száma. Könnyen belátható, hogy a (B^n-1) értéket - bármely számrendszerben - az ***n*** db. ***legnagyobb számjegyből*** álló szám adja, míg a B^n értékét – a legalacsonyabb helyértéktől kezdve – ***n*** db. ***legkisebb számjegyből***, és az ***n+1.*** helyen az ***eggyel*** nagyobb számjegyből álló szám adja.

Pl. n=5 esetén:

	(B^n-1)	B^n
decimális számoknál:	99999_d	100000_d
bináris számoknál	11111_b	100000_b
hexadecimális számoknál:	$FFFFF_h$	100000_h

Az első meghatározás szerintit nevezik ***(B-1)***-es, míg a másodikat ***B***-s komplemens -nek. A B a számrendszer alapszáma (radix).

A ***Z*** szám ***(B-1)***-es komplemens –ét $\overline{\overline{Z}}$- al, míg a ***B***-s komplemens –ét $\overline{Z}$ -al jelöljük. A kiegészítők – definíció szerinti - kiszámítása különbség-képzéssel történik. A számítás algoritmusa:

$$\overline{\overline{Z}} = (B^n - 1) - Z$$
$$\overline{Z} = B^n - Z$$

A választott számrendszer alapján beszélhetünk:

- a decimális számoknál ***kilences***-, illetve ***tízes-***, a
- a bináris számoknál ***egyes***-, és ***kettes***-,

komplemens –ről. (Más alapszám esetén az elnevezés hasonlóan adható meg.)

Az átszámítást – a kivonáson kívül – más eljárásokkal is elvégezhetjük. Előbb vezessük be a ***számjegy - komplemens*** fogalmát, amely az adott számjegy kiegészítő értéke a legnagyobb számjegyhez.

A Z szám (B-1)-es komplemens –ét megkapjuk, ha mindegyik helyértéken az adott számjegy kiegészítőjét írjuk:

$Z = 356_d$	$\overline{\overline{Z}} = 643_d$
$Z = 100110_b$	$\overline{\overline{Z}} = 011001_b$
$Z = 3A2B_h$	$\overline{\overline{Z}} = C5D4_h$

A Z szám B-s komplemens –ét kétféle módon is megkaphatjuk, ha figyelembe vesszük, hogy a kétféle kiegészítő ***különbsége*** – bármilyen B értéknél – ***1***, mivel $B^n - (B^n - 1) = 1$.

a. Képezzük a Z szám (B-1)-es komplemens –ét, és hozzáadunk 1-et.

$$\overline{Z} = \overline{\overline{Z}} + 1$$

$Z = 356_d$ $\quad \overline{\overline{Z}} + 1 = 643_d + 1 = 644_d$

$Z = 100110_b$ $\quad \overline{\overline{Z}} + 1 = 011001_b + 1 = 011010_b$

$Z = 3A2B_h$ $\quad \overline{\overline{Z}} + 1 = C5D4_h + 1 = C5D5_h$

b. A legkisebb helyértéktől kezdve a 0-kat leírjuk, az első „értékes" számjegy helyére a számjegy-kiegészítő + 1 értéket, míg a további számjegyek helyére, pedig azok kiegészítőjét írjuk.

$Z = 356_d$ $\quad \overline{Z} = 644_d$

$Z = 100110_b$ $\quad \overline{Z} = 011010_b$

$Z = 3A2B_h$ $\quad \overline{Z} = C5D5_h$

A ***digitális*** számítógépek bináris számokkal végeznek aritmetikai műveleteket . A ***negatív*** előjelű számoknál a ***kettes - komplemens*** használata gyorsabb műveletvégzést tesz lehetővé.

- A különböző számrendszerek közötti ***átszámítás***

A műszaki gyakorlatban leggyakrabban a ***decimális***, ***bináris***, és a ***hexadecimális*** számrendszereket használják. A következőkben röviden áttekintjük az átszámítások algoritmusát. Az ***emberek*** számára legfontosabb a ***decimális*** forma, mivel minden közérdekű számleírás ebben a formában történik. A számok ***gépi*** tárolása, és az azokkal végzett műveletek szinte kizárólag ***bináris*** rendszerben történik.

Általánosan az egyes számrendszerek közötti váltást (átszámítást) az új rendszer ***alap-számával*** történő ***sorozatos osztással*** végezhetjük el.

Először a ***decimális*** – ***bináris*** átalakítást ismételjük át. Számítsuk ki a 107_d érték bináris megfelelőjét.

107	**1**	2^0
53	**1**	2^1
26	**0**	2^2
13	**1**	2^3
6	**0**	2^4
3	**1**	2^5
1	**1**	2^6
0		

Tehát $\mathbf{107_d = 1101011_b}$

Gyakran van szükség a ***bináris*** – ***hexadecimális*** átszámításra is, mivel a számítástechnikai megjelenítés legtöbbször – a ***kisebb helyfoglalás*** érdekében – a bináris helyett a hexadecimális alakot használja.

Az átalakításnál 16 –al történő sorozatos osztást oly módon végezhetjük, hogy a bináris szám – legkisebb helyértékétől kezdődő – ***négy-négy*** számjegye helyett írjuk be a megfelelő ***hexadecimális*** számjegyet. Számítsuk át az előző példa értékét hexadecimális alakra.

$$\mathbf{107_d = 0110|1011_b = 6B_h}$$
$$\mathbf{6 \quad B}$$

Az utóbbi átszámítás – a leírtak szerint - könnyen elvégezhető fejben is. Csupán a hexadecimális számjegyek bináris megfelelőjét kell kiszámítani, vagy megjegyezni.

⇒ ***Számábrázolási (számírási) formák***

Az előzőekben csak a szám leírásának változatairól adtunk – a teljesség igénye nélkül – ismétlő áttekintést.

A műszaki, és egyéb gyakorlatban is legtöbbször ***különböző előjelű*** mennyiségek mérőszámait kell felírni, és azokkal műveletet végezni. A következőkben tömören – a teljesség igénye nélkül – összefoglaljuk azokat az ***előjegyes számleírási*** (számábrázolási) formákat, amelyeket a számításainkban használunk.

- ***Előjeles abszolút-értékes*** ábrázolás

A számleírás ilyen formáját használjuk a hétköznapi gyakorlatban a nyomtatott, és egyéb dokumentumokban. A szám ***pozitív***, vagy ***negatív*** voltát nem számjeggyel, hanem a +, vagy a – ***írásjellel*** adjuk meg a szám előtt. A szám értékét mindkét esetben ***abszolút-értékével*** írjuk. (Ez megfelel a számegyenesen jobbra-balra történő ábrázolásnak.)

- ***Előjegyes*** számábrázolás

A digitális számítógépek mind a ***számjegyeket***, mind a különböző ***írásjeleket*** kétértékű bitekkel tárolják. Az írásjelek ***kódolt*** formája ***8 bitet*** foglal le. A helytakarékosság, vala-

mint egyszerűbb műveletvégzési célból is, az ***előjelet*** is ***egy bittel*** – az ***előjegy*** –el – adják meg. A műszaki gyakorlatban ***0*** a ***pozitív***, az ***1*** pedig a ***negatív*** szám előjegy -e

Így beszélünk az ***előjegyes*** – számábrázolásról. A számrész megadási módja szerint megkülönböztetünk:

- előjegyes ***abszolút - értékes***, valamint
- előjegyes ***komplemens*** -es

formákat.

Az ***abszolút-értékes*** leírás tulajdonképpen az ***előjeles*** ábrázolás gépi változata. Ilyen alakú számokkal a műveletvégzés viszonylag összetett algoritmus szerint végezhető. A kijelölt, és az elvégzendő művelet (összeadás, vagy kivonás) a tényezők előjeleitől is függ. A tényleges műveletvégzés előtt döntés sorozatot kell végezni.

A ***komplemens*** –es ábrázolásoknál a ***pozitív*** számokat a számrész ***abszolút – értékével***, míg a ***negatív*** számokat, pedig a számrész valamelyik ***kiegészítőjével*** (komplemens-ével) adjuk meg.

pl. Írjuk fel a $+107_d$, illetve a -107_d számokat a különböző számábrázolási formában!

Előjeles decimális	**107**	**-107**
Előjegyes abszolutérték -es	**0 1101011**	**1 1101011**
1-es komplemens -ű	**0 1101011**	**1 0010100**
2-es komplemens -ű	**0 1101011**	**1 0010101**

⇒ ***Számok normál alakja***

A ***műszaki*** gyakorlatban, főleg a számítógépek széleskörű elterjedése előtt a különböző számítási műveletek elvégzését könnyítette az a számok ***normál alakban*** történt meg-adása.

A normál alak ***két részben*** adja meg a számot, mégpedig a ***számrészben***, és az ***exponenciális*** részben.

A számrészben a ***törtvessző előtt*** csak ***egyetlen*** – a legnagyobb helyértékű – ***számjegyet*** írjuk, míg a többi számjegy ***törtrészként*** szerepel. Utána kell leírni az exponenciális

részt, mint szorzó tényezőt, amely az ***alapszám*** (B) ***n*** -ik hatvány. Az ***n*** kitevő határozza meg, hogy a az ábrázolt szám milyen ***nagyságrendű***.

8.Példa. **$3{,}1023 * 10^4 = 3\ 1023$**

$5{,}234 * 10^{-2} = 0{,}05234$

$25\ 90{,}12 = 2{,}59012 * 10^3$

Az alap és a normál alakok közötti átírás a példákból egyértelmű. Az n kitevő formailag azt a számot jelenti, amennyivel a tizedes-vesszőt ***jobbra***, vagy ***balra*** kell vinni. Az irányt a ***kitevő előjele*** adja.

⇒ ***Bináris számok lebegőpontos (float) alakja***

A skalár számok ***lebegőpontos*** (float) ábrázolása, és tárolása - az IEEE-754 sz. szabványnak megfelelően - ***4 bájtban (32 bit)*** történik.

Az ábrázolási forma, a ***normál alakú*** számábrázolásnak a ***bináris*** számrendszerben történő alkalmazása.

A lebegőpontos szám két része az aktuális számot megadó un. ***mantissza***, és a nagyságrendet megadó kitevő, vagy másképp ***exponent***.

A ***kitevő*** 8 bites, amely 0 – 255 közötti érték adható meg. A ***kettes komplemens -***ű ábrázolás – az érték ***127***-el történő ***eltolása*** - lehetővé teszi, hogy negatív kitevőjű értéket is lehessen megadni, és ezzel a +128 és - 127 az értékkészlet szélső értékei.

A ***szám*** 24 biten fejezhető ki, de ebből ténylegesen csak ***23***-at, a tört-vesszőt követő részt tartalmazza a ***mantissza***. A normál alakú számábrázolásban csak egy, a 0-tól különböző számjegy lehet az egész részben. A bináris számoknál ez az 1, amit nem fontos megadni, mivel ez minden számnál azonos. Így lehet a 24 bites számot 23 biten megadni.

A négy bájtban – 32 biten - ábrázolt szám ***legnagyobb helyértékű*** bit a szám ***előjegy*** -e (signum).

A leírtak szerint tárolt lebegőpontos szám felépítése az alábbi:

SEEE EEEE EMMM MMMM MMMM MMMM MMMM MMMM

ahol:

S az előjegy bit, amely 0 értéke a pozitív, az 1, pedig a negatív számot jelzi,

E a 8 bites kitevő,

M a 23 bites mantissza.

Példa:

A **-12,5** értékű decimális szám lebegőpontos ábrázolásban **0xC1480000** lesz (a 32 bit helyett a rövidebb hexadecimális formát írtuk).

Értelmezzük az ábrázolási elv ismeretében az adott számot.

Forma **SEEEEEEE EMMMMMMM MMMMMMMM MMMMMMMM**

Bináris **11000001 01001000 00000000 00000000**

Hex.dec. **C1 48 00 00**

A legnagyobb helyértékű bit (**S**) 1, tehát a szám negatív.

Az következő nyolc bit (E-k) **10000010** a kitevőt adja, ha ebből levonjuk az eltolást, a 127-t. A binárisan leírt exponens decimális értéke 130, amelyből levonva 127-t 3-at kapunk, amely a tényleges kitevő.

Az utolsó 23 bit a mantissa (M-ek):

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

Mivel ez csak a törtvessző utáni rész, ezért még hozzá kell írnunk az egész-részt, vagyis 1-t. Az így kapott érték:

1.1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

amelyet szorozni kell $\mathbf{2^3}$ -al. Ekkor kapjuk meg a lebegőpontosan felírt szám ***abszolút-értékét***:

$$\mathbf{1100.1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0_b}$$

A szám egész része: 1100b binárisan, és átszámítva

$$\mathbf{(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 12}$$

decimális érték.

A törtvesszőt követő bináris rész: .100…b , átszámítva

$$\mathbf{(1 \times 2\text{-}1) + (0 \times 2\text{-}2) + (0 \times 2\text{-}3) + \ldots = 0.5}$$

decimális érték.

A két rész összege, és az előjel adja a lebegőpontosan ábrázolt szám decimális értékét. Tehát igazoltuk, hogy **0xC1480000** a **-12.5** szám lebegőpontos (float) formájú megadása.

⇒ ***Kódolt decimális számok***

A tízes számrendszerbeli számok közvetlenleírása, tárolása kódolt változatban is történhet. Miután a számítógépekben csak kétértékű elemi információk (bit-ek) tárolhatók, ezért a tíz számjegy csak ***több bit***-ből álló ***kód***-al helyettesíthető (írható le).

A tízes számrendszer számjegyeinek megadásához legkevesebb ***4 bitből*** álló kód szükséges, mivel ***3*** bittel csak ***8 érték*** különböztethető meg, viszont tíz értéket kell megkülönböztetnünk. A ***4 bites*** bináris kód viszont ***16*** különböző ***információt*** hordozhat, ezért ***10 értékhez*** rendelik a ***decimális számjegyeket***, és ***hat*** értéket ***nem*** használnak.

A 4 bites összerendelés, vagy más néven kódolás – az alábbi táblázatban bemutatott - három változatát használják.

BCD

	8	4	2	1
0	***0***	***0***	***0***	***0***
1	***0***	***0***	***0***	***1***
2	***0***	***0***	***1***	***0***
3	***0***	***0***	***1***	***1***
4	***0***	***1***	***0***	***0***
5	***0***	***1***	***0***	***1***
6	***0***	***1***	***1***	***0***
7	***0***	***1***	***1***	***1***
8	***1***	***0***	***0***	***0***
9	***1***	***0***	***0***	***1***
Nem használt	1	0	1	0
	1	0	1	1
	1	1	0	0
	1	1	0	1
	1	1	1	0
	1	1	1	1

Aiken

	2	4	2	1
0	***0***	***0***	***0***	***0***
1	***0***	***0***	***0***	***1***
2	***0***	***0***	***1***	***0***
3	***0***	***0***	***1***	***1***
4	***0***	***1***	***0***	***0***
Nem használt	0	1	0	1
	0	1	1	0
	0	1	1	1
	1	0	0	0
	1	0	0	1
	1	0	1	0
5	***1***	***0***	***1***	***1***
6	***1***	***1***	***0***	***0***
7	***1***	***1***	***0***	***1***
8	***1***	***1***	***1***	***0***
9	***1***	***1***	***1***	***1***

3 többletes (Stibitz)

	(8 4 2 1)-3			
Nem	0	0	0	0
	0	0	0	1
	0	0	1	0
0	***0***	***0***	***1***	***1***
1	***0***	***1***	***0***	***0***
2	***0***	***1***	***0***	***1***
3	***0***	***1***	***1***	***0***
4	***0***	***1***	***1***	***1***
5	***1***	***0***	***0***	***0***
6	***1***	***0***	***0***	***1***
7	***1***	***0***	***1***	***0***
8	***1***	***0***	***1***	***1***
9	***1***	***1***	***0***	***0***
haszn	1	1	0	1
	1	1	1	0
	1	1	1	1

A bemutatott kódok közül a ***BCD***-kód (Binary Coded Decimal), és az ***Aiken***-kód súlyozottak, ami azt jelenti, hogy az egyes bitek értéke ***2 hatványaival*** kifejezhető. A

Stibitz-kód ***eltolt-súlyozású***, ami azt jelenti, hogy a kód ***bináris*** értéke ***3-al több*** mint a hozzá rendelt ***decimális*** érték. Az utóbbi két kód szimmetrikus felépítésű, ugyanis a szaggatott vonaltól, - mint szimmetria tengelytől – egyenlő távolságra lévő kódok egymás 1-es kiegészítői. Ez a tulajdonság felhasználható hibajelzésre.

A ***BCD-kódot*** a ***számítástechnikában*** használják tízes számrendszerben történő szám-ábrázoláshoz, illetve számoláshoz. Az egy számjegyet leíró 4 bit-et ***dekád***-nak nevezzük. A 8 bites ***bájt***-ban ***két dekád*** írható, vagyis ***0 – 99*** decimális értéket tárolhat.

Pl. $\mathbf{38_d = 0011\ 1000_{BCD}}$

A mikroprocesszorok többségének utasításkészlete lehetővé teszi a BCD számokkal való számolást is. Erre a 2. féléves tananyagban térünk vissza.

Több bites kódokkal is leírhatók a decimális számjegyek. Itt csak az ***öt-bites*** un. ***Johnsson*** - kódot mutatjuk be.

0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	1	1
4	0	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	0
7	1	1	1	0	0
8	1	1	0	0	0
9	1	0	0	0	0
0	0	0	0	0	0

Az öt bit ***32*** érték kifejezését tenné lehetővé. Ebben az esetben csak ***tízhez*** rendeltünk ***értékes információt***. A kód tehát ***redundáns***. A további 22 érték ***hibajelzésre***, esetleg ***hibajavításra*** is használható.

A ***hibajelző***, és ***javító*** kódokkal – az ***információ-elmélet*** egy külön ága - a ***kódolás-elmélet*** foglalkozik részletesen. Ide tartoznak a különböző ***tömörítési***, ***titkosítási*** és ***visszafejtési*** stb. eljárások kidolgozása, algoritmizálása.

⇒ ***Aritmetikai műveletek algoritmusai***

A megismert számábrázolási formák közül a mikroprocesszoros rendszerekben (mik-rogép - ekben) általában a ***bináris kettes komplemens*** - ű változatot használják. Miután

az aritmetikai műveletek az összeadás, és a kivonás műveleteire vezethető vissza, ezért itt egy példán keresztül vizsgáljuk meg e két művelet elvégzésének szabályait.

Vegyük a 96, és a 43 abszolút-értékű számok közötti műveleteket. Mind az összeadásnál, mind pedig a kivonásnál négy-négy műveletet kell elvégeznünk.

Az adott számok bináris kettes komplemens –ü értékei:

96	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**-96**	**1**	**0**	**1**	**0**	**0**	**0**	**0**
43	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**-43**	**1**	**1**	**0**	**1**	**0**	**1**	**0**

Szaggatott vonallal az előjegy – bitet határoltuk el.

Az elvégzendő műveleteket láthatók az alábbiakban. Mindkét műveletnél helyértékenként - a legkisebb helyérték kivételével – három számjegyet (bit -et) adunk össze, illetve vonunk ki. Ezek a két szám ***azonos helyértékű számjegyei*** (bit -jei), illetve az előző helyértéken keletkező ***átvitel*** (***Cy*** Carry), illetve ***áthozat*** (***Bw*** Borrow) bitek. Az utóbbi értékeket a negyedik sorba - egy kissé eltolva - írtuk, jelezve ezzel a helyérték-váltást.

A ***kettes komplemens***-ű ábrázolású számok esetében ***mindig*** a ***kijelölt*** műveletet – az összeadást, vagy a kivonást – kell végezni úgy, hogy az ***előjegy bitet*** is ***számbitként*** kezeljük. A példában félkövér számmal jelöltük az ***utolsó számjegynél***, és az ***előjegy bitnél*** keletkezett ***átvitel / áthozat*** biteket.

Összeadás

96	0	1	1	0	0	0	0	96	0	1	1	0	0	0	0
+ 43	0	0	1	0	1	0	1	+ -43	1	1	0	1	0	1	0
139	1	0	0	0	1	0	1	53	0	0	1	1	0	1	0
	0	***1***	1	0	0	0	0		***1***	***1***	0	0	0	0	0

-96	1	0	1	0	0	0	0	-96	1	0	1	0	0	0	0
+ 43	0	0	1	0	1	0	1	+ -43	1	1	0	1	0	1	0
-53	1	1	0	0	1	0	1	-139	0	1	1	1	0	1	0
	0	***0***	1	0	0	0	0		***1***	***0***	0	0	0	0	0

Kivonás

	96		0	1	1	0	0	0	0		96		0	1	1	0	0	0	0	0
-	43		0	0	1	0	1	0	1	1	-	-43	1	1	0	1	0	1	0	1
	53		0	0	1	1	0	1	0	1		139	1	0	0	0	1	0	1	1
		0	***0***	1	1	1	1	1	1				***1***	***0***	0	1	1	1	1	1

	-96		1	0	1	0	0	0	0	0	-96		1	0	1	0	0	0	0	0
-	43		0	0	1	0	1	0	1	1	-	-43	1	1	0	1	0	1	0	1
	-139		0	1	1	1	0	1	0	1		-53	1	1	0	0	1	0	1	1
		0	***1***	1	1	1	1	1	1				***1***	***1***	0	1	1	1	1	1

A példák elvégzése után megállapíthatjuk a szabályokat:

- Mindkét műveletnél az ***eredményt*** is ***kettes komplemens***-ű formában kapjuk. A ***pozitív*** szám ***előjegy*** -e ***0***, és a számrész ***abszolút*** értékű, míg a ***negatív*** szám ***előjegy*** -e ***1***, és a számrész a szám ***kettes komplemens*** -e.

- ***Hibátlan*** eredményt kapunk, ha a két utolsó ***átvitel/áthozat*** bit (az utolsó számjegynél, illetve az előjegynél) 00, vagy 11.

- ***Hibás*** az eredmény, ha csak az egyik helyen keletkezik ***átvitel/áthozat*** bit. Ilyenkor ***aritmetikai túlcsordulás*** van. Ez azt jelenti, hogy a keletkezett eredmény nem fér el a számrésznek fenntartott helyen, (kicsi a kapacitás) vagyis az eredmény nagyobb, mint a 7 bittel megadható legnagyobb érték. A hiba a tároló-hely ***kapacitásának növelésével*** küszöbölhető ki.

- A két túlcsordulás-bit ***XOR*** (moduló 2) műveletének eredménye az un. ***Overflow*** –bit (***OF***), amit ***aritmetikai túlcsordulás*** bitnek is neveznek. Minden mikroprocesszor un. ***Status***, vagy ***Flag*** bitjei között szerepel az OF bit.

A ***kettes komplemens*** –ű számábrázolás előnye tehát az, hogy ***gyorsabb*** a művelet-végzés. A negatív számok abszolút értékre való konvertálását, illetve fordítva csak az adat ***kiíratásánál***, illetve ***bevitelénél*** kell elvégezni. A műveletek a gépen belül gyorsabban hajthatók végre.