Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Irányítástechnika és Informatika Tanszék

Dr. Pilászy György

Hardveralapok
03. előadás

(Logikai érték, logikai algebra)

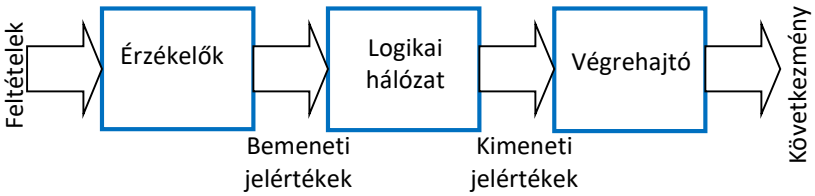
Lektorálta: Dr. Horváth Tamás

Minden jog fenntartva. Jelen könyvet, illetve annak részleteit a szerzők írásbeli engedélye nélkül tilos reprodukálni, adatrögzítő rendszerben tárolni, bármilyen formában vagy eszközzel elektronikus vagy más módon közölni.

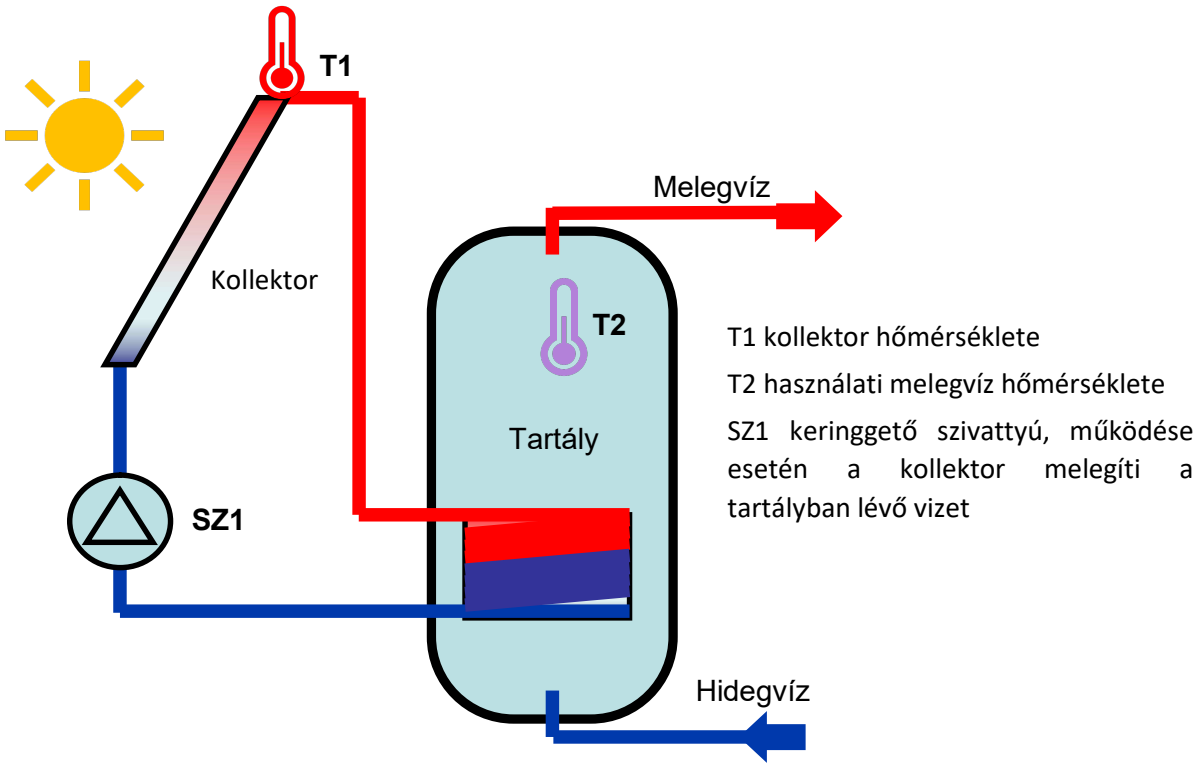
Logikai érték

A hétköznapi életben gyakran hozunk döntéseket. Egy állításról mondhatjuk azt, hogy igaz, de azt is mondhatjuk, hogy hamis. Az állítással kapcsolatosan tett „igaz” vagy „hamis” elnevezést az állítás logikai értékének nevezzük. Ezt a két diszkrét értéket a továbbiakban „1” illetve „0” értékkel fogjuk jelölni.

A logikai hálózatok világában az a feladatunk, hogy a környezetben tapasztalható mennyiségeket logikai értékekre képezzük le. A logikai értékek alapján feltételeket fogalmazunk meg, majd a feltételek alapján létrehozzuk a kimeneti logikai értékét. A logikai feladat megoldását logikai döntésnek is nevezhetjük. A feltételekhez rendelt következményeket ezen döntés alapján hozzuk létre.



Az elmondottak szemléltetésére tekintsünk egy szép irányítástechnikai feladatot. Legyen a példánk egy használati melegvíz készítő rendszer vezérlése. A rendszer vázlatát mutatja a következő ábra. Napsütéses időben a Kollektorban lévő fagyálló folyadék felmelegszik, majd egy úgynevezett hőcserélőn keresztül tudja melegíteni a tartályban lévő vizet, ha az SZ1 jelű szivattyú működik.



© 2020, Pilászy György

A felhasználó igénye az, hogy a tartályban legalább 40°C-os víz legyen. Ez alapján megfogalmazhatunk egy logikai feltételt a melegvíz készítésére vonatkozóan: T2<40°C (1)

Természetesen, ha nem süt eléggé a nap, akkor nem tudunk vizet melegíteni. A gépész tervezőnk azt javasolta, hogy legalább 60°C-os vízzel kell melegítenünk, ezért egy újabb logikai feltételt fogalmazhatunk meg a kollektorra vonatkozóan: T1>60°C (2)

Ezen két feltétel alapján hozhatunk egy döntést, mely alapján az SZ1 jelű szivattyút elindíthatjuk vagy leállíthatjuk. Ezeket a döntéseket táblázatos formában is megadhatjuk. A következő táblázat [1] a lehetséges feltételeket és a hozzájuk tartozó következményeket foglalja össze. A táblázatból kiolvashatjuk, hogy az SZ1 szivattyú csak akkor működik, ha mindkét feltétel egyidejűleg teljesül. Az egyes sorokhoz rövid szöveges megjegyzést is írtunk a könnyebb megértés érdekében.

Feltételek		**Következmény**	**Megjegyzés**
T2<40°C	**T1>60°C**	**SZ1**	
hamis	hamis	szivattyú leállítva	Van elég melegvíz és nem süt a nap
hamis	igaz	szivattyú leállítva	Van elég melegvíz és süt a nap
igaz	hamis	szivattyú leállítva	Nincs elég melegvíz, de nem süt a nap
igaz	igaz	szivattyú elindítva	Nincs elég melegvíz és süt a nap

Feltételek és következmények összerendelése [1. táblázat]

A példából érzékelhetjük, hogy milyen összetett feladat a logikai érték előállítása, de ha ez sikerült, akkor viszonylag egyszerűen fogalmazhatjuk meg a következményekre vonatkozó előírásainkat. A környezetünkben általában valamilyen fizikai jellemző alapján kell logikai értéket előállítanunk. Ez a jellemző lehet nyomás, hőmérséklet, ellenállás, feszültség, áram, fényintenzitás stb. A fenti példában a hőmérséklet értékének összehasonlításaként (komparátor) állíthatjuk elő a logikai értékeket. A továbbiakban kifejezetten a villamos feszültségen alapuló logikai értékekkel, úgynevezett feszültség logikákkal foglalkozunk.

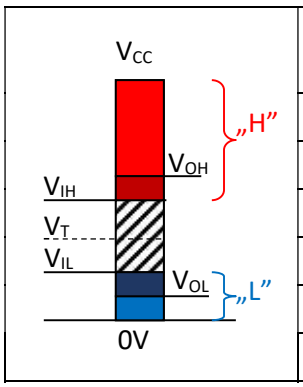
Feszültséglogika jelszintjei

Feszültséglogika alkalmazásakor az igaz/hamis logikai értékekhez jól mérhető feszültség tartományokat rendelünk. Ezek a tartományok két, jól elkülöníthető sávra oszthatók. Az építőelem működéséhez szükséges tápfeszültséget általában VCC (ritkábban VDD) és GND (ritkábban VSS) elnevezésekkel jelölik, ahol a VCC a pozitív, míg GND (ground) a 0V feszültségű pont. A GND pont referenciának is tekinthető, a logikai építőelem összes bemenetének és kimenetének a feszültségszintjeit ehhez viszonyítva adják meg. A tápfeszültséghez közelebb eső tartományt gyakran nevezik „H”-szintnek vagy „Magas” logikai szintnek, míg a 0V-hoz közelebb eső sávot „L”-szintnek, vagy „Alacsony” szintnek. Az L és H sávok között meghatározhatunk egy úgynevezett „tiltott” sávot is, ahol a váltások során fellépő tranziens időszakot kivéve, tartósan nem lehet a jel feszültségének értéke. A tiltott sávban található egy V_T -vel jelölt küszöbérték (threshold), mely a logikai építőelem belső úgynevezett billenési feszültségének névleges értékét jelöli. Ha egy 0V-ból induló folyamatosan növekvő feszültséget kapcsolunk a logikai építőelem bemenetére, akkor ezen érték „környékén” következik be a logikai értékek közötti váltás valahol a tiltott sávon belül. A későbbiekben ismertetésre kerülő logikai építőelemek bemenetére és kimenetére kis mértékben eltérő tartományokat definiálnak. A „Magas” logikai szinthez, az építő elem bemenetén érzékelhető feszültségre a V_{IH} (input, high), a kimenetén a V_{OH} (output, high) küszöbértéket adják meg. Az „Alacsony” logikai szinthez, az építő elem bemenetén érzékelhető feszültségre a V_{IL} (input, low), a kimenetén a V_{OL} (output, low) küszöbértéket

© 2020, Pilászy György

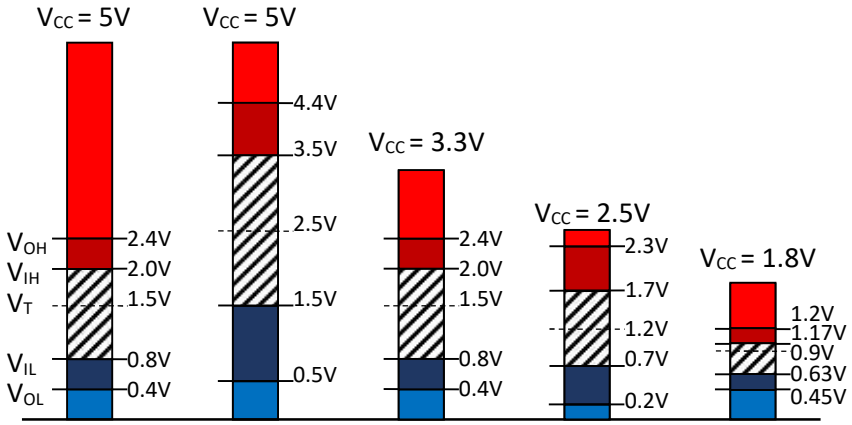
adják meg. Ez a gyakorlatban annyit jelent, hogy az építőelem gyártója garantálja, hogy az előírt működési feltételek betartása esetén a „Magas” szinthez tartozó kimeneti feszültség V_{OH}…V_{CC} közé esik, míg „Alacsony” szintű kimenet esetén 0V…V_{OL} közé. Ugyanígy az építő elem a bemenetére kapcsolt 0V…V_{IL} közé eső feszültséget „Alacsony” szintnek, míg a V_{IH}…V_{CC} közé eső feszültséget „Magas” szintnek érzékeli.

Az alábbi táblázatban [2] a katalógusokban általában használt jelölések mellett megadjuk néhány elterjedt logika feszültség szintjeit. Megjegyezzük, hogy a tényleges gyártmányok küszöbértékei ettől kis mértékben eltérhetnek, ezért tervezéskor mindig ajánlott ellenőrizni a pontos értékeket. A táblázatban szereplő rövidítések jelentése: TTL- Tranzisztor-Tranzisztor logika, LVTTL- Low Voltage TTL, CMOS- Complementary Metal Oxide Semiconductor.



Logika elnevezése	TTL (5V)	CMOS (5V)	LVTTL (3.3V)	CMOS (2.5V)	CMOS (1.8V)
V_{CC}	5V	5V	3.3V	2.5V	1.8V
V_{OH}	2.4V	4.44V	2.4V	2.3V	1.2V
V_{IH}	2.0V	3.5V	2.0V	1.7V	1.17V
V_{IL}	0.8V	1.5V	0.8V	0.7V	0.63V
V_{OL}	0.4V	0.5V	0.4V	0.2V	0.45V
V_T	1.5V	2.5V	1.5V	1.2V	0.9V

Különböző feszültség logikák jeltartományai [2. táblázat]



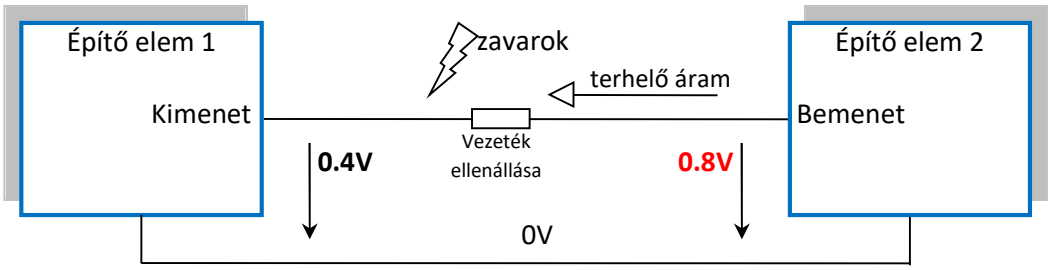
Különböző logikai szintek ábrázolása [1. ábra]

Zajtartalék (noise margin)

Egyazon áramköri családon belül is megfigyelhetjük, hogy az építőelemek bemenetére és kimenetére eltérő feszültségszint-tartományokat specifikálnak. Ennek a gyakorlatban a zavarvédelem szempontjából van kiemelt jelentősége. A logikai áramkörök tervezésekor fontos szempont volt, hogy ugyanazon áramköri családon belüli építőelemek be- és kimeneteit közvetlenül (huzalozással) lehessen összekapcsolni más illesztő áramkörök nélkül. Ha egy építőelem kimenetét összekapcsoljuk egy másik építő elem bemenetével, akkor az összekötő vezetéken különböző nagyságú terhelő áramok folynak, ennek hatására feszültség különbség lehet a vezeték két vége között (vezetett zavarok). Ugyancsak problémát okozhatnak a rádiófrekvenciás sugárzásokból eredő zavarok, melyek a vezetékben zavaró feszültségeket indukálhatnak, amely szintén feszültség különbséget okoz a vezeték két vége között

© 2020, Pilászy György

(sugárzott zavarok). Amennyiben a zavaró jel nagysága alatta marad a V_{OH}-V_{IH}; V_{IL}-V_{OL} feszültségek különbségénél, úgy az nem okoz hibát a logikai hálózat működésében. Ezt nevezzük zajtartaléknak. Például, ha egy TTL építőelem kimenetén közvetlenül 0.4V feszültség jelenik meg, amelyhez 0.4V zavaró jel adódik hozzá a vezetéken folyó zavaró áramok miatt, akkor a vezeték másik végén lévő bemeneten 0.8V feszültség lesz, melyet még „L” szintnek érzékel az áramkör. Ha a zavarójel mértéke ennél nagyobb, akkor a bemeneten érzékelhető feszültség belecsúszik a „tiltott” sávba, ahol nem tudjuk pontosan megmondani, hogy „L” vagy „H” szintnek fogja érzékelni az építőelemünk. Ez az előírt tartományon kívül eső jel bizonytalan működéshez vezethet.



Logikai építő elemek összekapcsolása [2. ábra]

Összekapcsolhatóság

A hardvertervezői gyakorlatban előforduló probléma, hogy különböző feszültségről működő logikai építőelemeket kell összekapcsolni. A fenti ábrát és táblázatot figyelmesen áttekintve megállapíthatjuk, hogy a zajtartalék rovására sok esetben ez lehetséges. Amennyiben az építőelemek tápfeszültsége is különbözik, úgy a túl magas logikai szint jelenléte is okozhat problémát, mert a legtöbb építőelem nem viseli el károsodás nélkül a bemenetén fennálló „túl magas” feszültségszintet. A lehetséges összekapcsolhatóságokat szemlélteti a következő táblázat [3]. A táblázatban *-al jelölt esetekben az összekapcsolás csak akkor lehetséges, ha a bemenet tolerálja a tápfeszültséget meghaladó V_{IH} logikai szintet. Ha a megengedettől eltérő feszültségű logikákat kell összekapcsolni, akkor azt közvetlenül nem, csak megfelelő szintillesztő áramkörön keresztül lehet üzembiztosan megvalósítani. A logikai építőelem gyártók kínálatában megtalálhatók ezek a szintillesztő (level translator, level shifter) áramkörök.

bemenet / **kimenet**	**5V TTL**	**5V CMOS**	**3.3V TTL**	**2.5V CMOS**	**1.8V CMOS**
5V TTL	igen	nem	igen*	igen*	igen*
5V CMOS	igen	igen	igen*	igen*	igen*
3.3V TTL	igen	nem	igen	igen*	igen*
2.5V CMOS	igen	nem	igen	igen	igen*
1.8V CMOS	nem	nem	nem	nem	igen

Különböző feszültségű logikák összekapcsolhatósága [3. táblázat]

© 2020, Pilászy György

Pozitív és negatív logika

A katalógusok általában nem igaz/hamis, hanem H és L vagyis „magas” és „alacsony” logikai szinteket specifikálnak. A tényleges összerendelés így a tervező feladata lesz. Amennyiben a H-szinthez rendeljük az „1” logikai értéket és az L-szinthez a „0” értéket, úgy **pozitív logikáról** beszélünk. Egyes esetekben célszerű lehet a fordított hozzárendelés, ekkor **negatív logikáról** beszélünk.

Logikai érték	**jelölés**	**Pozitív logika**	**Negatív logika**
igaz	1	H	L
hamis	0	L	H

Feszültség szintek és logikai értékek összerendelése [4. táblázat]

Feszültség logika építőelemeinek kimenetei

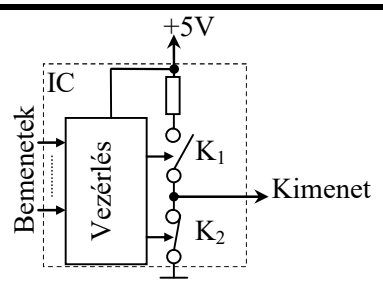
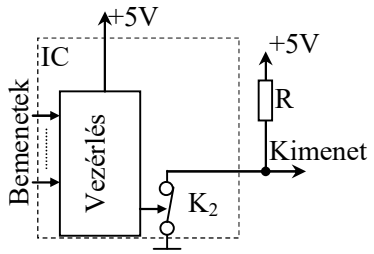
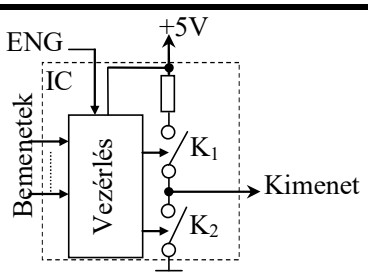
Az előzőekben megismert feszültségszintek biztosításához úgy tervezték meg a logikai építőelemeket, hogy a működésnek megfelelő „H” vagy „L” szinthez rendelt feszültség jelenjen meg az építőelem kivezetésén. Anélkül, hogy az áramköri ismereteket részleteznénk a következőkben nagyon röviden bemutatjuk a TTL áramkörökben alkalmazott három kimenet típust, a Totem-Pole (TP), a nyitott kollektor (OC) és a háromállapotú (3st) kimenetet.

A **Totem-Pole** kimenet tulajdonképpen két elektronikus kapcsoló elemből áll, a vezérlésnek kell biztosítania, hogy a kimenet a tápfeszültségre (H-szint) vagy a földpontra (L szint) kapcsolódjon. A TP kimenet alkalmas több kapubemenet meghajtására. Mivel a belső kapcsolók egyike mindig zárt (közel rövidzár), ezért a TP kimenetek egymással nem köthetők össze, mert két ellentétes értékű kimenetnél a belső kapcsoló elemek tönkre mehetnek.

A **nyitott kollektoros** kimenet egyetlen kapcsoló elemet tartalmaz, amely a kimenetet a földpontra kapcsolja (L-szint). A magas logikai szintet megfelelően méretezett külső felhúzó ellenállással kell biztosítani. Az ilyen kapukimenetek feltétel nélkül összeköthetők, ilyenkor az egyes kimenetek között egy úgynevezett huzalozott függvénykapcsolat jön létre.

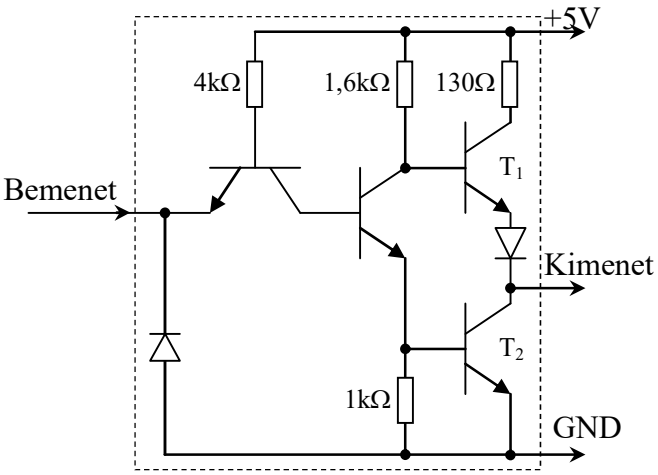
A **háromállapotú** kimenetek a H és L szinteken kívül rendelkeznek egy nagyimpedanciás állapottal is, amelyben kimeneti pont és az áramkör közötti kapcsolat megszakítottnak tekinthető. Az ilyen tulajdonságú áramkörök kimenetei összeköthetők egymással, de csak akkor, ha megfelelő vezérléssel garantáljuk, hogy egyszerre legfeljebb csak egy kapukimenet aktív. A harmadik állapot előállításához egy külön tiltó / engedélyező bemenet tartozik.

© 2020, Pilászy György

Totem-Pole	Nyitott kollektoros	Háromállapotú
		
L-szint: K1: nyit, K2: zár **H**-szint: K1: zár, K2: nyit	**L**-szint: K2: zár **H**-szint: K2: nyit	**L**-szint: K1: nyit, K2: zár **H**-szint: K1: zár, K2: nyit **3. állapot**: K1 nyit, K2 nyit

Logikai építőelemek kimeneteinek modellezése [5. táblázat]

Az integrált áramkörökben a kapcsoló elemeket természetesen nem mechanikus kapcsolókkal, hanem félvezetőből kialakított elektronikus kapcsoló elemekkel (tranzisztorokkal) valósítják meg. Az alábbi ábrán egy Totem-Pole kimenetű TTL inverter (SN7404) belső rajza látható [3].



Inverter belső kapcsolási rajza [3. ábra]

Megjegyzés:

A kimeneti modellben a K_1 és K_2 kapcsolókat a T_1 illetve T_2 tranzisztorok valósítják meg.

© 2020, Pilászy György

Logikai változó, logikai függvény

Az előző fejezetben megismertük a logikai érték fogalmát, konkrét példákat láttunk a logikai értékek megvalósítására. Könnyen beláthatjuk, hogy nincs értelme ezekkel a fizikai paraméterekkel közvetlenül aritmetikai műveleteket (pl.: összeadás, kivonás) végezni, hiszen az eredményként kapott paraméter értékeknek a logikai feladat szempontjából nincs jelentésük, ráadásul a feszültségszinteknek üzemszerűen az előírt tartományokon belül kell maradniuk a helyes működés érdekében.

A logikai műveletekben a logikai érték viselkedése hasonló az aritmetikai műveletekben alkalmazott számértékekhez, azonban amíg az aritmetikai értékek végtelen sokasága létezik, addig a logikai érték esetében csak véges számú érték fordul elő. A továbbiakban kétértékű logikával foglalkozunk, az „igaz” értéket 1-el, a „hamis” értéket 0-val fogjuk jelölni.

Az aritmetikai kifejezésekben használt betűjelek aritmetikai változókat jelölnek, melyekhez hasonlóan a logikai kifejezésekben logikai változókat fogunk használni. A logikai építőelemek, logikai hálózatok bemeneti és kimeneti pontjait is egy-egy logikai változóval jelölhetjük. A logikai változók kizárólag 0 vagy 1 értéket vehetnek fel. Azokat a függvényeket, amelyekben a függő és a független változók is logikai változók, logikai függvényeknek nevezzük.

Logikai alapműveletek

Az előző pontban már láttuk, hogy a logikai változók között nincs értelme aritmetikai műveletek használatának. A következőkben definiáljuk a logikai változókon értelmezett logikai alapműveleteket. Mivel a függő és a független változók is mindössze két értéket vehetnek fel, ezért a továbbiakban táblázatos formában adjuk meg az egyes műveleteket.

Logikai szorzás

Konjunkció, ÉS kapcsolat, (AND), jele: ·

A	B	F = A·B	értelmezés
0	0	0	0·0 = 0
0	1	0	0·1 = 0
1	0	0	1·0 = 0
1	1	1	1·1 = 1

Az ÉS kapcsolat végeredmény akkor „1” értékű, ha az összes benne szereplő független változó „1” értékű. Az ÉS művelet kommutatív tulajdonságú, a független változók sorrendje felcserélhető. Érdemes megfigyelni, hogy az ÉS művelet szabálya formálisan megegyezik az aritmetikai szorzással, azonban a benne szereplő változóknak logikai jelentésük van.

© 2020, Pilászy György

Logikai összeadás

Diszjunkció, VAGY kapcsolat (OR), jele: +

A	B	F = A+B	értelmezés
0	0	0	0+0 = 0
0	1	1	0+1 = 1
1	0	1	1+0 = 1
1	1	1	1+1 = 1

A VAGY kapcsolat eredménye akkor „1” értékű, ha legalább egy független változó logikai értéke „1”.
A VAGY művelet kommutatív tulajdonságú, a független változók sorrendje felcserélhető.

Tagadás

Invertálás, negálás, komplementálás, ellentett képzés, jele: felülvonás

A	$F = \overline{A}$	értelmezés
0	1	$\overline{0} = 1$
1	0	$\overline{1} = 0$

Az invertálás a logikai értékhez az ellentettjét rendeli hozzá. Páros számú invertálás alkalmazása esetén a kiinduló logikai értéket kapjuk vissza. A későbbiekben, ha szeretnénk egy invertálás nélküli logikai változóról vagy annak invertáltjáról nyilatkozni akkor a „ponált” és „negált” kifejezéseket fogjuk használni.

Logikai algebrai kifejezés

Az előzőekben megismert logikai értékek (mint konstansok), logikai változók és logikai alapműveletek együttesen matematikai értelemben logikai algebrának vagy Boole algebrának nevezhetők. Mindezek együttes felhasználásával az alábbi azonosságokhoz jutunk. Az azonosságot $\equiv$ szimbólummal jelöljük és így olvassuk: „azonosan egyenlő”.

$$A \cdot 0 \equiv 0 \qquad A + 0 \equiv A$$
$$A \cdot 1 \equiv A \qquad A + 1 \equiv 1$$
$$A \cdot A \equiv A \qquad A + A \equiv A$$
$$A \cdot \overline{A} \equiv 0 \qquad A + \overline{A} \equiv 1$$
$$\overline{\overline{A}} \equiv A$$

A Boole algebra kommutativitását (felcserélhetőség) írja le az alábbi két összefüggés:

$$A + B \equiv B + A$$
$$A \cdot B \equiv B \cdot A$$

© 2020, Pilászy György

A Boole algebra asszociatív (csoportosítható) tulajdonságát szemléltetik az alábbi összefüggések:

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

A disztributivitás tulajdonságait pedig az alábbi összefüggések mutatják:

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

Számunkra a későbbiekben fontos lesz még az alábbi két összefüggés, melyet **De-Morgan** azonosságnak is hívnak és az ÉS - VAGY műveletek közötti dualitást fejezik ki.

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

Ezeknek az azonosságoknak a helyességéről magunk is meggyőződhetünk, ha az A és B logikai változók helyébe behelyettesítjük a „0”, illetve az „1” logikai értékeket.

Logikai függvények

Azokat a függvényeket, amelyekben mind a függő, mind a független változó logikai változó, logikai függvénynek nevezzük. Minden kimeneti változó értéke a bemeneti változók értékétől függ. Ha n bemenete és m kimenete van egy logikai rendszernek, akkor az m számú n változós logikai függvénykapcsolattal megadható.

A következőkben határozzuk meg az összes két változós logikai függvényt. A korábbiakban láttuk, hogy a logikai algebra diszkrét értékeinek köszönhetően véges sok variációja létezik a bemeneti változóknak ezért táblázatos formában is megadhatjuk az összes lehetséges kombinációt. Jelölje A és B a két bemeneti változót. A logikai változók kétértékűsége miatt összesen 2^2=4 különböző kombinációnk lehetséges, ezért készítsünk egy négy soros táblázatot. A kimenetre vonatkozó előírásokat az egyes oszlopokban jelenítsük meg. Egy adott logikai függvény esetén összesen 4 cella kitöltése szükséges. Figyelembe véve az összes variációs lehetőséget 2^4=16 különböző logikai függvényt tudunk megadni. Ezeket a következő táblázatban f_0…f_{15} jelöléssel adtuk meg. Az alábbi táblázatos megadási formát szokás a függvény igazságtáblázatának is hívni.

független változó kombinációk		Függvényértékek															
A	**B**	**f_0**	**f_1**	**f_2**	**f_3**	**f_4**	**f_5**	**f_6**	**f_7**	**f_8**	**f_9**	**f_{10}**	**f_{11}**	**f_{12}**	**f_{13}**	**f_{14}**	**f_{15}**
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Az összes lehetséges kétváltozós logikai függvény igazságtáblázata

Nézzük meg a kapott f_0…f_{15} függvényeket, próbáljuk meg felírni a logikai függvények egy-egy algebrai alakját. Mivel mindössze két logikai érték létezik, ezért a függvény megadásához elegendő azokat a független változókombinációkat megadni, amelyekhez „1” függvényérték tartozik. Ugyanígy egyértelműen megadja a függvényt az összes „0” függvényértéket eredményező független változó kombinációk megadása.

© 2020, Pilászy György

A fenti táblázatban található f_0 és f_{15} függvények kimeneti értéke független az A,B logikai változóktól, ezek konstans értékek.

$$f_0(A,B) = 0\,,\;\; f_{15}(A,B) = \overline{f_0(A,B)} = 1$$

Az f_1 függvénykapcsolat csak akkor „1” értékű, ha minden változója „1” értékű. Ez megfelel a logikai szorzat definíciójával, vagyis

$$f_1(A,B) = A \cdot B$$

Az f_2 függvény akkor ad „1” értéket, ha A=„1” és B=„0” vagyis $\overline{B} = 1$. Ezt algebrai alakban így fogalmazhatjuk meg:

$$f_2(A,B) = A \cdot \overline{B}$$

Az f_3 függvény két különböző kombináció esetén ad „1” értéket, az A=„1”, B=„0” és az A=„1” B=„1” esetén. Ezt algebrai alakban a következőképpen írhatjuk:

$$f_3(A,B) = A \cdot \overline{B} + A \cdot B$$

A logikai algebra fentebb megismert szabályait alkalmazva ezt a kifejezést átalakíthatjuk az alábbiak szerint:

$$f_3(A,B) = A \cdot \overline{B} + A \cdot B = A \cdot (\overline{B} + B) = A$$

Vagyis megállapíthatjuk, hogy az f3 függvény független a B változótól.

$$f_3(A,B) = A$$

Az f_4 függvény akkor ad „1” értéket, ha A=„0” és B=„1” vagyis $\overline{A} = 1$. Ezt algebrai alakban így fogalmazhatjuk meg:

$$f_4(A,B) = \overline{A} \cdot B$$

Az f_3 függvényhez hasonlóan az f_5 független az A változótól, ezért az alábbi alakban írhatjuk fel:

$$f_5(A,B) = B$$

Az f6 függvény akkor ad „1” értéket, ha a két változó közül csak az egyik „1” értékű. Ezt úgy is megfogalmazhatjuk, hogy akkor „1” értékű, ha különbözik a két változó, ezért ezt a függvénykapcsolatot szokták antivalencia vagy „KIZÁRÓ VAGY” kapcsolatnak is hívni. Ezt a függvénykapcsolatot külön műveleti jellel is szokták jelölni: $\oplus$.

$$f_6(A,B) = \overline{A} \cdot B + A \cdot \overline{B} = A \oplus B$$

Az f_7 függvény megegyezik a VAGY műveletnél tanultakkal, ezért a következő alapban írhatjuk fel:

$$f_7(A,B) = A + B$$

Ugyanerre az alakra juthatunk, ha az f_7 függvény „1” értéket adó kombinációit egymással VAGY kapcsolatba hozzuk és alkalmazzuk a megismert azonosságokat:

$$f_7(A,B) = \overline{A} \cdot B + A \cdot \overline{B} + A \cdot B = (A \cdot \overline{B} + A \cdot B) + (\overline{A} \cdot B + A \cdot B) =$$
$$= (A \cdot (\overline{B} + B)) + (B \cdot (\overline{A} + A)) = A + B$$

A fenti átalakítás során kihasználtuk, hogy az AB logikai szorzatot többször is hozzávehetjük a kifejezéshez.

© 2020, Pilászy György

Az f_8…f_{15} függvények értékeit megnézve megfigyelhetjük, hogy az f_0..f_7 függvények ellentettjeit adják. Az f8 függvény csak akkor ad „1” értéket, ha egyidejűleg az A és a B változó is „0” értékű ezért az alábbi alakban írhatjuk fel:

$$f_8(A,B) = \overline{f_7(A,B)} = \overline{A} \cdot \overline{B}$$

Az f_9 függvény az antivalencia ellentettje. Ez akkor ad „1” értéket, ha a két logikai változó azonos értékű, ezért szokás ekvivalencia függvénynek is hívni. Az ekvivalencia műveletet külön jellel is szokás jelölni: ⊗

$$f_9(A,B) = \overline{f_6(A,B)} = \overline{A} \cdot \overline{B} + A \cdot B = A \otimes B$$

A következő függvények esetében használjuk ki azt a felismerésünket, hogy az f_{10}…f_{14} függvények az f_5…f_1 függvények ellentettjeiként előállíthatók.

$$f_{10}(A,B) = \overline{f_5(A,B)} = \overline{B}$$

Alkalmazzuk a De-Morgan összefüggést a korábban meghatározott algebrai alakok ellentettjeire.

$$f_{11}(A,B) = \overline{f_4(A,B)} = \overline{\overline{A} \cdot B} = A + \overline{B}$$

$$f_{12}(A,B) = \overline{f_3(A,B)} = \overline{A}$$

$$f_{13}(A,B) = \overline{f_2(A,B)} = \overline{A \cdot \overline{B}} = \overline{A} + B$$

$$f_{14}(A,B) = \overline{f_1(A,B)} = \overline{A \cdot B} = \overline{A} + \overline{B}$$

Logikai függvények megadási módjai

A kombinációs hálózatok tervezésének első lépése, hogy egy szöveges specifikáció (informális leírás) alapján formálisan, későbbi tervezési módszerek alkalmazásához alkalmas módon specifikáljuk a kívánt logikai hálózatot. Az előző pontban megismertük a logikai függvények **igazságtáblázatát** és láttunk példákat a függvények **algebrai alakjára** is. A függvények algebrai alakjainak felírásánál VAGY kapcsolatba hoztuk azokat a szorzatokat, amelyek esetén a függvény „1” értéket ad. A szorzatokban ponáltan (invertálás nélkül) szerepelt az „1” értékű és negáltan a „0” értékű változó.

Az igazságtáblázat egyértelmű megadását jelenti a logikai függvényeknek, azonban nagy számú független változó alkalmazása esetén a hatalmas mérete miatt nem minden esetben célszerű ilyen módon megadni a hálózatot. Az algebrai alak jó lehet, azonban több, látszólag nem azonos algebrai alakja is lehet ugyanazon függvénynek, ami nehézkessé teszi a különböző megoldások ellenőrzését, összehasonlítását. Vizsgáljuk meg például a következő két algebrai alakot:

$$F_1 = B \cdot \overline{C} + A \cdot \overline{B} \qquad F_2 = (A+B) \cdot (\overline{B} + \overline{C})$$

Ránézésre különbözőek, hiszen az F_1 két szorzat VAGY kapcsolatából áll, az F_2 viszont két VAGY kapcsolat szorzata. F1 alakját diszjunktív algebrai alaknak, F2 alakját konjunktív algebrai alaknak nevezzük.

Végezzük el az F2 kifejezésben a zárójelek felbontását:

$$F_2 = (A+B) \cdot (\overline{B} + \overline{C}) = A\overline{B} + A\overline{C} + B\overline{B} + B\overline{C}$$

A kapott kifejezést bővítsük az alábbiak szerint:

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

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

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

© 2020, Pilászy György

Ezt behelyettesítve és elhagyva az azonos tagokat a következő eredményre jutunk:

$$F_2 = A \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C} = A \cdot \overline{B} \cdot (C + \overline{C}) + B \cdot \overline{C} \cdot (A + \overline{A}) = A \cdot \overline{B} + B \cdot \overline{C}$$

Vagyis algebrai átalakításokkal sikerült az F2-t F1-el megegyező algebrai alakra hoznunk, tehát igazoltuk, hogy a két kifejezés ugyanazt az igazságtáblát adja meg.

Adjuk meg most ennek az F1=F2=F logikai függvénynek az igazságtáblázatát. Állítsunk elő szisztematikusan olyan szabályos algebrai alakokat, amelyek egyértelműen megadják az F függvényt.

A	B	C	F
0	0	0	0
0	0	1	0
0	**1**	**0**	**1**
0	1	1	0
1	**0**	**0**	**1**
1	**0**	**1**	**1**
1	**1**	**0**	**1**
1	1	1	0

A fenti azonosság igazolása során alkalmazott bővítést felhasználva és kihasználva a logikai függvények kétértékű tulajdonságát, állítsuk elő az F függvényt logikai szorzatok összegeként úgy, hogy minden szorzatban szerepeljen a függvény összes független változója.

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

Figyeljük meg, hogy ennek a logikai összegben minden egyes szorzat pontosan egy bemeneti kombinációra ad „1” értéket. Mivel a fenti igazságtáblában összesen négy ilyen hely van, ezért az összegben négy szorzat került VAGY kapcsolatba. Úgy is mondhatjuk, hogy a függvényt „1”-esenként raktuk össze. Ezt az algebrai alakot **diszjunktív kanonikus algebrai alaknak** nevezzük.

<u>A diszjunktív kanonikus algebrai alak tulajdonságai:</u>

- mindegyik szorzat olyan független változó kombinációt képvisel, ahol a függvényérték „1”
- minden szorzatban az összes független változó szerepel,
- pontáltan szerepelnek azok a független változók, melyekhez „1” bemeneti érték tartozik,
- negáltan szerepelnek azok a független változók, melyekhez „0” bemeneti érték tartozik.

A fenti tulajdonságok alapján megállapíthatjuk, hogy a diszjunktív kanonikus algebrai alak speciális elemi logikai függvények összege. Ezeket az elemi függvényeket mintermeknek nevezzük.

A mintermek jelölésére vezessük be az alábbi jelölést:

$$m_i^n$$

ahol n jelöli a független változók darabszámát, i pedig az illető mintermnek megfelelő változó kombináció decimális értéke (minterm index). A minterm index képzése úgy történik, hogy a mintermhez tartozó ponált változókat „1”, a negált változókat „0” bináris számjeggyel helyettesítjük, majd a kapott n bites bináris számot átváltjuk decimális alakba. A későbbiekkel összhangban, ha A, B, C,…. jelöléssel látjuk el a független változókat, akkor mindig az A változót tekintjük a legmagasabb helyi értékűnek.

© 2020, Pilászy György

Ezek alapján a fenti F függvény a következőképpen írható fel:

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

$$\overline{A} \cdot B \cdot \overline{C} \rightarrow 010 \rightarrow m_2^3$$

$$A \cdot \overline{B} \cdot \overline{C} \rightarrow 100 \rightarrow m_4^3$$

$$A \cdot \overline{B} \cdot C \rightarrow 101 \rightarrow m_5^3$$

$$A \cdot B \cdot \overline{C} \rightarrow 110 \rightarrow m_6^3$$

$$F(A,B,C) = m_2^3 + m_4^3 + m_5^3 + m_6^3$$

A fenti mintermes alakot egyszerűbben is leírhatjuk, ha a mintermek közötti logikai összegeket egy ∑ betűvel helyettesítjük, felsőindexként megadjuk a független változók darabszámát, majd felsoroljuk a függvényhez tartozó minterm indexeket.

$$F(A,B,C) = m_2^3 + m_4^3 + m_5^3 + m_6^3 = \sum^{3}(2,4,5,6)$$

Ezt a megadási módot nevezzük minterm indexes megadásnak.

Vegyük elő újra a kiindulási függvényünk igazságtáblázatát.

A	B	C	F
0	**0**	**0**	**0**
0	**0**	**1**	**0**
0	1	0	1
0	**1**	**1**	**0**
1	0	0	1
1	0	1	1
1	1	0	1
1	**1**	**1**	**0**

Próbáljuk most előállítani az F logikai függvényt az F=”0” értékeinek felhasználásával.

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

Figyeljük meg, hogy ennek a logikai szorzatban minden egyes összeg pontosan egy bemeneti kombinációra ad „0” értéket. Mivel a fenti igazságtáblában összesen négy ilyen hely van, ezért az szorzatban négy összeg került ÉS kapcsolatba. Úgy is mondhatjuk, hogy a függvényt „0”-nként raktuk össze. Ezt az algebrai alakot **konjunktív kanonikus algebrai alaknak** nevezzük.

<u>A konjunktív kanonikus algebrai alak tulajdonságai:</u>

- mindegyik összeg olyan független változó kombinációt képvisel, ahol a függvényérték „0”
- minden összegben az összes független változó szerepel,
- pontáltan szerepelnek azok a független változók, melyekhez „0” bemeneti érték tartozik,
- negáltan szerepelnek azok a független változók, melyekhez „1” bemeneti érték tartozik.

A fenti tulajdonságok alapján megállapíthatjuk, hogy a konjunktív kanonikus algebrai alak speciális elemi logikai függvények összege. Ezeket az elemi függvényeket maxtermeknek nevezzük.

© 2020, Pilászy György

A maxtermek jelölésére vezessük be az alábbi jelölést:

$$M_i^n$$

ahol n jelöli a független változók darabszámát, i pedig az illető maxtermek megfelelő változó kombináció decimális értéke (maxterm index). A maxterm index képzése úgy történik, hogy a maxtermhez tartozó algebrai alakban szereplő ponált változókat „1”, a negált változókat „0” bináris számjeggyel helyettesítjük, majd a kapott n bites bináris számot átváltjuk decimális alakba. A későbbiekkel összhangban, ha A, B, C,…. jelöléssel látjuk el a független változókat, akkor mindig az A változót tekintjük a legmagasabb helyi értékűnek. Ezek alapján a fenti F függvény a következőképpen írható fel:

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

$$(A+B+C) \rightarrow 111 \rightarrow M_7^3$$

$$(A+B+\overline{C}) \rightarrow 110 \rightarrow M_6^3$$

$$(A+\overline{B}+\overline{C}) \rightarrow 100 \rightarrow M_4^3$$

$$(\overline{A}+\overline{B}+\overline{C}) \rightarrow 000 \rightarrow M_0^3$$

$$F(A,B,C) = M_0^3 \cdot M_4^3 \cdot M_6^3 \cdot M_7^3$$

A fenti maxtermes alakot egyszerűbben is leírhatjuk, ha a maxtermek közötti logikai szorzatokat egy Π betűvel helyettesítjük, felsőindexként megadjuk a független változók darabszámát, majd felsoroljuk a függvényhez tartozó maxterm indexeket.

$$F(A,B,C) = M_0^3 \cdot M_4^3 \cdot M_6^3 \cdot M_7^3 = \prod^3 (0,4,6,7)$$

Ha elég bátornak érezzük magunkat, akkor megpróbálhatjuk algebrai átalakításokkal is előállítani a diszjunktív kanonikus algebrai alakból a konjunktív kanonikus algebrai alakot. Ehhez vegyük elő az eredeti F függvény negáltjának diszjunktív kanonikus algebrai alakját. Ebben azok a mintermek szerepelnek, melyek az eredeti F függvényben nem szerepeltek, hiszen most az eredeti „0” helyek lettek az „1”-esek.

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

Alkalmazzuk a De-Morgan szabályt a VAGY műveletekre:

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

Alkalmazzuk a De-Morgan szabályt a zárójelben lévő kifejezésekre:

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

végeredményként megkaptuk az előbb meghatározott konjunktív kanonikus algebrai alakot. A kapott eredményünket alkalmazhatjuk a minterm indexes-maxterm indexes alakok közvetlen átalakítására is.

$$F(A,B,C) = \sum^3 (2,4,5,6)$$

© 2020, Pilászy György

Első lépésben állítsuk elő az F függvény minterm indexeinek komplemens halmazát, ezek lesznek ugyanis az F negált függvény minterm indexei:

$$\overline{F(A,B,C)} = \sum^{3}(0,1,3,7)$$

Alkalmazzuk a De-Morgan szabályt az indexekre és a műveleti jelre. Az indexek „negálása” bitenkénti invertálást jelent, ami jelen esetben a 7-ből kivonást jelenti a három változó miatt (7-0→7; 7-1→6; 7-3→4; 7-7→0).

$$\overline{\overline{F(A,B,C)}} = \prod^{3}(7,6,4,0)$$

Megfigyelhetjük, hogy a korábbi módon meghatározott maxterm indexeket kaptuk. Természetesen a maxterm indexekből kiindulva ugyanilyen módon visszakaphatjuk a függvény minterm indexeit is.

$$F(A,B,C) = \prod^{3}(7,6,4,0) \rightarrow \overline{F(A,B,C)} = \prod^{3}(1,2,3,5)$$

$$\overline{\overline{F(A,B,C)}} = \sum^{3}(6,5,4,2)$$

Az eddigiek alapján tehát az alábbi logikai függvény megadási módokat ismertük meg:

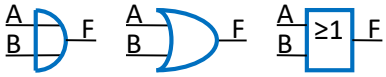
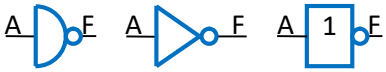
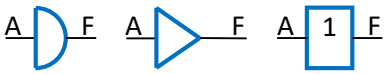
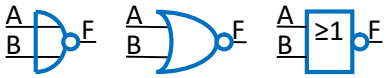
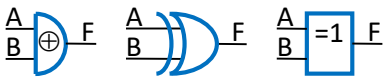
- Igazságtáblázat
- Algebrai alak
- Diszjunktív kanonikus algebrai alak
- Konjunktív kanonikus algebrai alak
- Mintermes alak
- Minterm indexes alak
- Maxtermes alak
- Maxterm indexes alak

A következőkben még egy gyakran használt ábrázolási módot ismertetünk.

© 2020, Pilászy György

Logikai függvény megadása elvi logikai rajzzal

Az integrált áramkörök megjelenésével lehetőség nyílt nagy tömegben elérhető áron gyártani elemi kombinációs hálózatokat. Ezeket az elemi műveleteket megvalósító vagy elemi műveletekből felépített hálózatokat logikai kapuknak nevezzük. Ezeknek a kapuáramköröknek több elterjedt rajzjele van, melyekből a leggyakrabban használtakat a következő táblázatban foglalunk össze.

Megnevezés	Rajzjelek	Logikai függvény
ÉS, AND		$F = A \cdot B$
VAGY, OR		$F = A + B$
Inverter		$F = \overline{A}$
Buffer		$F = A$
NEM-ÉS, NAND		$F = \overline{A \cdot B}$
NEM-VAGY, NOR		$F = \overline{A + B}$
KIZÁRÓVAGY, XOR		$F = \overline{A}B + A\overline{B} = A \oplus B$

Elvi logikai rajzjelek

Megjegyezzük, hogy a VAGY kapuk rajzjelében korábban ’1’ bejegyzés szerepelt, ez később módosították ’≥1’ feliratra. Ez utóbbi felirat úgy is értelmezhető, hogy a kapu kimenete akkor „1”, ha legalább egy bemenete „1” értékű. Ehhez hasonlóan az XOR műveletben szereplő ’=1’ bejegyzést tekinthetjük úgy, hogy akkor „1” a kapu kimente, ha pontosan egy bemenete „1” értékű. A fenti táblázatban kétbemenetű függvényeket ábrázoltunk. Az elvi logikai rajzokon elvileg tetszőlegesen sok bemenete lehet egy szimbólumnak, azonban a kereskedelemben is kapható kapuáramkörökből csak néhány létezik.

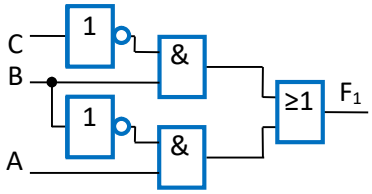
© 2020, Pilászy György

Az első három sorban a jól ismert alapműveleteket láthatjuk. A negyedik sorban szereplő BUFFER áramkör látszólag vezetékként funkcionál, azonban a gyakorlatban jelerősítő szereppel bír. A NEM-ÉS, NEM-VAGY kapuáramkörök nem alapműveletek, azonban a későbbiekben látni fogjuk, hogy ezekkel is bármilyen logikai alapművelet megvalósítható.

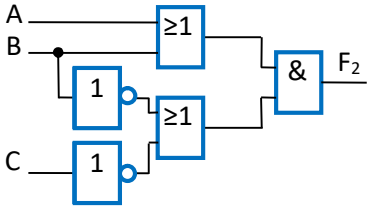
A kapuáramkörök és rajzjeleik felhasználásával is megadhatjuk a kombinációs hálózatokat. Ez az elvi logikai rajz már konkrét építőelemeknek is megfeleltethető, így a kapcsolási rajzok alapját képezi. Az építőelemek bal oldalán találhatók a logikai bemenetek, míg jobb oldalon az építő elem kimenete. A be- és kimenetek közötti jeltovábbítást vezetékek segítségével valósítjuk meg. Ezeket az összeköttetéseket folytonos vonallal ábrázoljuk az elvi logikai rajzon. Az egymást keresztező vonalakat nem tekintjük összekötöttnek, csak ha külön csomóponttal jelöltük.

Ábrázoljuk a korábban megismert F_1 és F_2 logikai függvényt a most megismert szimbólumok felhasználásával.

$$F_1 = B \cdot \overline{C} + A \cdot \overline{B}$$



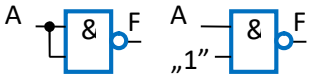
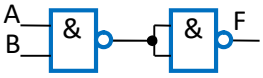
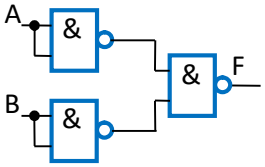
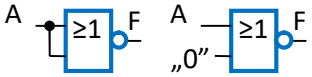
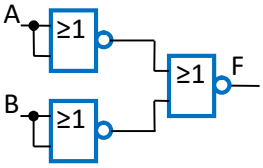
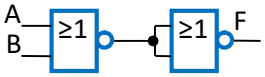
$$F_2 = (A + B) \cdot (\overline{B} + \overline{C})$$



Az F1 és F2 logikai függvények ábrázolása elvi logikai rajzzal

Invertált kimenetű kapuk használata

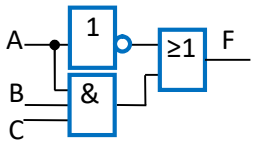
A logikai építőelemek ismertetése kapcsán találkoztunk a NEM-ÉS (NAND) és NEM-VAGY (NOR) kapukkal. A következőkben nézzük meg, hogy kizárólag egyféle kapu felhasználásával hogyan lehet előállítani bármelyik tanult alapműveletet.

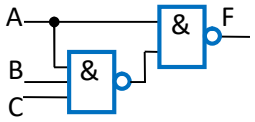
Alapművelet Megvalósítás	**Invertálás**	**ÉS**	**VAGY**
NAND			
NOR			

Logikai alapműveletek megvalósítása NAND, illetve NOR kapukkal

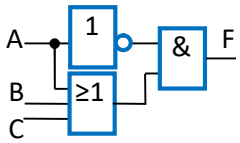
© 2020, Pilászy György

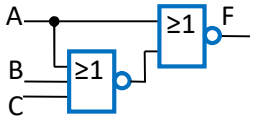
Ábrázoljuk ÉS-VAGY, majd kizárólag NAND kapukkal a következő diszjunktív algebrai alakot: $F(A,B,C)=\overline{A}+ABC$. A megoldás algebrai levezetéséhez a De-Morgan szabályt alkalmazzuk a diszjunktív alakban szereplő VAGY műveletre: $F(A,B,C)=\overline{A}+ABC=\overline{\overline{(\overline{A})}\cdot\overline{(ABC)}}=\overline{A\cdot\overline{ABC}}$. Figyeljük meg, hogy a második szinten lévő NAND kapu VAGY műveletet végez. A második szintre „direktbe” csatlakozó független változó invertálódik.





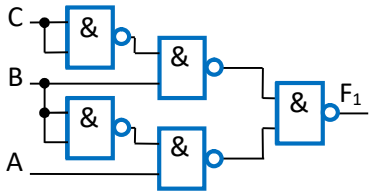
Ábrázoljuk VAGY-ÉS, majd kizárólag NOR kapukkal a következő konjunktív algebrai alakot: $F(A,B,C)=\overline{A}\cdot(A+B+C)$. A megoldás algebrai levezetéséhez a De-Morgan szabályt alkalmazzuk a konjunktív alakban szereplő ÉS műveletre: $F(A,B,C)=(\overline{A})\cdot(A+B+C)=\overline{\overline{(\overline{A})}+\overline{(A+B+C)}}=\overline{A+\overline{(A+B+C)}}$. Figyeljük meg, hogy a második szinten lévő NOR kapu ÉS műveletet végez. A második szintre „direktbe” csatlakozó független változó invertálódik.



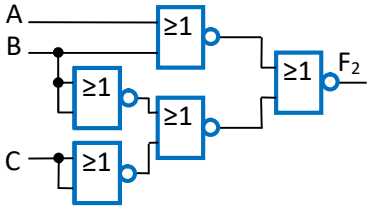


Ábrázoljuk a korábban megismert F_1 függvényt kizárólag NAND, az F_2 logikai függvényt kizárólag NOR kapuk felhasználásával. A megoldás helyességének algebrai igazolásához alkalmazzuk a De-Morgan azonosságot az F1 függvény esetében a VAGY műveletre, az F2 függvény esetében pedig az ÉS műveletre:

$$F_1=B\cdot\overline{C}+A\cdot\overline{B}\rightarrow\overline{\overline{(B\cdot\overline{C})}\cdot\overline{(A\cdot\overline{B})}}$$



$$F_2=(A+B)\cdot(\overline{B}+\overline{C})\rightarrow\overline{\overline{(A+B)}+\overline{(\overline{B}+\overline{C})}}$$



Az F1 és F2 logikai függvények ábrázolása NAND illetve NOR kapukkal

Figyeljük meg a hasonlóságot az F1 függvény esetében az ÉS-VAGY kapus és a NAND kapus megoldás között. Láthatjuk, hogy az építőelemek közötti összekötések pontosan ugyanúgy alakulnak, csak a kapuk lettek kizárólag NAND kapukra cserélve. Ugyanígy viselkedik az F2 függvény VAGY-ÉS kapus megoldása és a kizárólag NOR kapus realizáció. Ezt a megfigyelésünket a későbbiekben fel fogjuk használni.

© 2020, Pilászy György

Közömbös bejegyzések

A logikai feladatok megfogalmazása, specifikálása során előfordulhat, hogy bizonyos bemeneti kombinációk nem fordulhatnak elő a hálózat bemenetén. Könnyen beláthatjuk, hogy a bemeneten fel nem lépő kombinációkra tetszőleges előírást tehetünk, hiszen a működés során sosem lép fel. Ezeket a bejegyzéseket **közömbös bejegyzéseknek** hívjuk és az igazságtáblázatban „-” bejegyzéssel jelöljük. Amennyiben szükséges, a számjegyes vagy algebrai felírásoknál ezeket a termeket is megjeleníthetjük, de elkülönítve a határozott bejegyzésektől. A közömbös bejegyzések gyakorlati hasznát a később ismertetett minimalizálási eljárások során fogjuk megmutatni, most csak annyit jegyzünk meg, hogy bizonyos esetekben ezek megfelelő értékű rögzítésével egyszerűbb függvényalak állítható elő.

A közömbös bejegyzést tartalmazó logikai hálózatot **nem teljesen specifikált logikai hálózatnak** nevezzük. Ha egyetlen közömbös bejegyzést sem tartalmaz, akkor a hálózatot **teljesen specifikált logikai hálózatnak** nevezzük.

A következőkben nézzünk egy háromváltozós példát nem teljesen specifikált logikai függvény megadására az eddig tanult megadási módokban.

Egy logikai hálózat bemenetére három kapcsoló csatlakozik (A,B,C). A hálózat kimenete F=1, ha az A kapcsoló „1” értéke mellett a B és C különböző értékű. A megadáskor vegyük figyelembe, hogy nem állhat mindhárom kapcsoló azonos állásban.

Adjuk meg az F függvény igazságtáblázatát, diszjunktív és konjunktív kanonikus algebrai alakját, mintermes és maxtermes alakját, valamint a minterm indexes és maxterm indexes alakját.

A függvény igazságtáblája:

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

Diszjunktív kanonikus algebrai alak:

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

Mintermes alak:

$$F(A,B,C) = m_5^3 + m_6^3 + (m_0^3 + m_7^3)$$

Minterm indexes alak:

$$F(A,B,C) = \sum^3 [(5,6) + (0,7)]$$

Konjunktív kanonikus algebrai alak:

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

Maxtermes alak:

$$F(A,B,C) = M_6^3 + M_5^3 + M_4^3 + M_3^3 + (M_7^3 + M_0^3)$$

Maxterm indexes alak:

$$F(A,B,C) = \prod^3 [(3,4,5,6)(0,7)]$$

A feladat leírásában vegyük észre, hogy a legutolsó mondat a közömbös bejegyzéseket határozza meg. Emiatt az igazságtábla legelső és legutolsó sorába közömbös bejegyzések kerültek. Az algebrai alakok felírásánál a közömbös bejegyzéseket elkülönítve, külön zárójelben tüntettük fel. A közömbös bejegyzéseket hasonlóan jelöltük a mintermes és maxtermes alakoknál is. Az indexes alakok megadásánál szintén külön zárójelek közé kerültek a közömbösöket jelölő indexek.

© 2020, Pilászy György