

I. modul: Kombinációs hálózatok

2. lecke: nevezetes kétváltozós függvények; implementáció kétszintű kapu-hálózatokkal; logikai függvények egyszerűsítési módszerei

Cél:

A lecke célja, hogy a hallgató megismerje a Boole-algebra nevezetes kétváltozós függvényeit, az ezekhez tartozó alapvető logikai kapu szimbólumokat, illetve megismerje a logikai függvények egyszerűsítési eljárásait és az ehhez tartozó fogalmakat

Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha

- ismeri a Boole-algebra nevezetes kétváltozós függvényeit
- ismeri az alapvető logikai kapukat és szimbólumaikat
- képes egy logikai függvény kétszintű kapuhálózatának realizációjára
- ismeri és alkalmazni tudja a logikai függvények algebrai és grafikus egyszerűsítési eljárásait

Időszükséglet:

A tananyag elsajátításához kb. 120 percre van szükség.

Kulcsfogalmak:

- nevezetes kétváltozós logikai függvény
- implikáns, prímimplikáns
- Quine-módszer
- Karnaugh tábla
- teljesen vagy nem teljesen specifikált kombinációs hálózat

Tevékenység: írja fel táblázatos formában a nevezetes kétváltozós függvényeket, és rajzolja fel az őket megvalósító kapu szimbólumokat! Valósítsa meg realizációs rajz elkészítésével a leckében szereplő függvények kapusintű logikai áramköreit!

2.1 Nevezetes kétváltozós logikai függvények

Kétváltozós (X_1, X_2) logikai függvények esetében léteznek kitüntetett, más néven nevezetes kétváltozós logikai függvények, melyek közül többet gyakran használunk logikai hálózatok építése során, mint kapu-áramköröket. Ezek értelmezéséhez nyújt segítséget a 2.1.a ábrán látható táblázat.

BEM. VÁLT.		FÜGGVÉNYÉRTÉKEK															
X_1	X_2	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

2.1.a ábra

Az összes lehetséges kétváltozós logikai függvény ($F_0 - F_{15}$)

A táblázatban szereplő függvények közül nevezetes kétváltozós logikai függvényeknek tekintik a következőket:

- **'0' (F_0) és '1' (F_{15}) generátor:**

Az F_0 értéke állandóan '0', nem érzékeny a bemenetekre. Negáltja az F_{15} , amelynek értéke állandóan '1'.

- **'ÉS' (F_1) és 'NEM-ÉS' (F_{14}) függvény, illetve kapu:**

Az F_1 és az F_{14} értékei éppen egymás negáltjai. Az F_1 nevezetes, csak akkor szolgáltat '1'-et, ha mindkét bemeneti változó értéke '1', az F_{14} pedig ekkor éppen '0'-t szolgáltat. Az előbbi neve 'ÉS', az utóbbié 'NEM-ÉS', röviden 'NÉS'.

- **'VAGY' (F_7) és 'NEM-VAGY' (F_8) függvény, illetve kapu:**

Az F_7 és az F_8 függvények ugyancsak egymás negáltjai. Az F_7 értéke '1', ha legalább ez egyik bemenet '1'. Az F_8 értéke éppen ilyenkor '0'. Az előbbi a 'VAGY', az utóbbi a 'NEM-VAGY' ('NVAGY') függvény, illetve kapu.

- **ANTIVALENCIA (F_6) és EKVIVALENCIA (F_9) függvény, illetve kapu:**

Ha a kétváltozós függvény csak abban az esetben szolgáltat '1'-et, ha a két bemenet különböző, akkor a függvény neve: ANTIVALENCIA, vagy KIZÁRÓ-VAGY (EXOR, XOR). Ez az F_6 függvény. Negáltja az F_9 , éppen ezekben az esetekben '0', és egyezés esetén '1'. Ez utóbbi neve EKVIVALENCIA, vagy KIZÁRÓ-NEM-VAGY (EXNOR, XNOR). Az ezeknek megfelelő kapu áramköröket igen gyakran használjuk. A KIZÁRÓ-VAGY művelet jelölésére gyakran használják a ' $\oplus$ ' szimbólumot.

2.1.1 Logikai függvények egyszerűsítése

A logikai függvények egyszerűsítésének célja egyrészt a realizáció során a logikai kapubemenetek, valamint a realizáláshoz felhasznált kapuk számának csökkentése, másrészt ezáltal a minél egyszerűbb és költséghatékonyabb áramköri megvalósítás elérése. A csupa NOR, illetve NAND kapus, valamint a multiplexeres megvalósítások pedig az SSI áramkörök (IC-k) darabszámát csökkentik.

Ebben a fejezetben azt a négy egyszerűsítési eljárást mutatjuk be, melyek a teljesen határozott logikai függvények esetében általánosságban alkalmazhatóak. Ezek a következők:

- *egyszerűsítés algebrai módszerrel*
- *Quine módszer*
- *Quine-McCluskey módszer*
- *Karnaugh táblás módszer*

2.1.1.1 Algebrai módszer

Egyszerűsítsük a bemutatott azonosságokat kihasználva a már megismert háromváltozós logikai függvényünket! A disztributivitást kifejező azonosságok arra mutatnak, hogy a klasszikusan kiemelésnek nevezett átalakítás itt is végrehajtható. A logikai összeg első két szorzat-termjéből és a második két szorzat-termjéből is kiemelhető egy-egy kétváltozós szorzat. Ezután felismerhetjük a zárójelben lévő összegek '1' értékét, valamint azt, hogy a '1'-el való beszorzást nem kell feltüntetni. Az eredmény kifejezés tehát azonos a kiindulással, de jóval egyszerűbb annál:

$$\begin{aligned} F(A, B, C) &= \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + ABC \\ &= \overline{A}B(\overline{C} + C) + A\overline{C}(\overline{B} + B) = \overline{A}B + A\overline{C} \end{aligned}$$

2.1.1.2 A Quine módszer

A Quine módszer alapja a függvény '1'-es értékéhez rendelt két minterm közös szorzótényezőinek oly módon történő kiemelése, hogy a zárójelben egy logikai változónak és negáltjának az összege maradjon, amely logikai összeg '1'.

Példa: adott a korábbi példánkban szereplő mintermes függvény leírás, melyhez az alábbi táblázatot készíthetjük.

<i>mintermek</i>	<i>I.</i>	<i>II.</i>	<i>III.</i>
2	$\overline{A}\overline{B}\overline{C} \checkmark$	(2, 3) $\overline{A}\overline{B}$	
3	$\overline{A}\overline{B}C \checkmark$	(2, 6) $\overline{B}\overline{C}$	
4	$\overline{A}B\overline{C} \checkmark$	(4, 6) $A\overline{C}$	
6	$AB\overline{C} \checkmark$		

2.1.1.2.a ábra[1]

A Quine-módszer oszlopai

Gondoljunk arra, hogyan hajtottuk végre az algebrai egyszerűsítést az

$$\overline{A}\overline{B}\overline{C}$$

és az

$$\overline{A}B\overline{C}$$

mintermek közös tényezőjének, az

$$\overline{A}B$$

kiemelésével. A zárójelen belül a

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

marad. Ezzel a két összevont minterm helyén a közös szorzó-tényező marad, a zárójelen belül maradó változó eltűnik. A Quine-módszer lényege ennek az eljárásnak a szisztematikus ismétlése mindaddig, amíg ilyen eltüntethető változó már nem marad. Ha

a változók száma 'n', először az 'n' tényezős minterm párokat vonjuk így össze 'n-1' tényezős termékké, azután a kiadódó 'n-1' tényezős term-párokat 'n-2' tényezős

termékké, azután a kiadódó 'n-2' tényezős term-párokat 'n-3' tényezős termékké, és így tovább.

Az 2.1.1.2.a ábrán mutatjuk a szisztematikus eljárást ismert példánkra. Azok a mintermek, amelyekhez a függvény '1'-et rendel, az 'I.'-el jelölt oszlopban láthatók, bináris értékeikkel megcímezve. A II. oszlopban az összevonható minterm-párok címei és az összevonások eredményei láthatók. Ha a II. oszlopban lennének összevonható kéttényezős szorzatok, akkor a harmadik oszlop nem lenne üres.

Figyeljük meg, a kiemeléssel összevonható termék sajátossága, hogy azok mindig csak egyetlen változóban különböznek egymástól. Azt mondjuk, hogy az ilyen termék ún. Hamming-féle távolsága egységnyi.

Miután nincs több oszlop, ahová új összevonásokat írhatnánk, az oszlopokban szereplő, további összevonásokba már nem bevonható termeket prímisszimplikánsoknak nevezzük. Példánk prímisszimplikánsai :

$$\overline{AB}, \quad \overline{BC}, \quad \overline{AC}$$

A Quine-módszer következő lépése a feltétlenül szükséges prímisszimplikánsok kiválasztása. Ezt a prímisszimplikánsok lefedési táblázatának segítségével végezhetjük el (2.1.1.2.b ábra). A prímisszimplikánsok kijelölik a táblázat sorait, az oszlopokat pedig azok a mintermek, amelyekhez a függvény '1'-et rendel. Ezután '*' karakterrel bejelöljük az egyes prímisszimplikánsok által lefedett mintermeket.

Prímisszimplikáns táblázat

	2	3	4	6
(2, 3) $\overline{AB}$	*	*		
(2, 6) $\overline{BC}$	*			*
(4, 6) $\overline{AC}$			*	*

2.1.1.2.b ábra[1]

Prímisszimplikánsok lefedési táblája

Nyilvánvaló például, hogy az

$$\overline{A}B$$

prímimplikáns az

$$\overline{A}B\overline{C}$$

és az

$$\overline{A}BC$$

mintermeket fed. A lefedési tábla megmutatja, elhagyhatunk-e úgy egy prímimplikánst, hogy a lefedendő mintermek mindegyike fedve marad.

Ha találunk ilyeneket, azokat elhagyhatjuk. Példánkban egyetlen redundáns, tehát elhagyható prímimplikáns a

$$B\overline{C}$$

A lényeges, elhagyhatatlan prímimplikánsok tehát :

$$\overline{A}B, \quad A\overline{C}$$

A végeredmény a lényeges prímimplikánsok logikai összege, tehát:

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

A fentiekben megismert néhány új fogalom összefoglalva:

- **prímimplikáns:**
olyan term, amelyből nem hagyható el több változó
- **megkülönböztetett minterm :**
olyan minterm, amelyet csak egy prímimplikáns fed le

- **lényeges prímiplikáns:**
olyan prímiplikáns, amely csak egy megkülönböztetett mintermet tartalmaz.

2.1.1.3 A Quine-McCluskey módszer

A Quine-McCluskey egyszerűsítési módszer részletes elemzése és bemutatása nem képezi e tárgy törzsanyagának részét, ezért röviden összefoglalva a következőket kell tudni a módszer lényegéről: ez az egyszerűsítési módszer a Quine eljárásból alakult ki, de a mintermek összevonhatóságát nem a bináris kódok közötti távolság, hanem a decimális értékek alapján vizsgálja.

Könnyen megfogalmazhatók ugyanis azok a kritériumok, amelyek fennállása esetén két minterm, illetve két term összevonható. Ennek a megközelítésnek nagy előnyei, hogy a 4-5-nél nagyobb bemeneti változó szám esetén is könnyen alkalmazható, és egyszerű a számítógépes megvalósítás. A Quine-McCluskey módszer a számítógépes logikai szintézis eljárások előfutárává vált a múlt század hatvanas éveiben.

A módszer lényegének összefoglalása az alábbi linken tanulmányozható:

https://en.wikipedia.org/wiki/Quine%E2%80%93McCluskey_algorithm

2.1.1.4 A Karnaugh táblás (grafikus) módszer

A Karnaugh-táblás minimalizálás lényegében a Quine-eljárás geometriai reprezentációja. A táblázat a nevét az elv kidolgozójának, Maurice Karnaugh amerikai fizikusnak a nevééről kapta.

Az egyszerűsítés elvégzéséhez szükséges táblázat megalkotásának elve a következő: egy 'n' változós függvény lehetséges mintermjeinek megfeleltetünk egy-egy négyzetet, és úgy helyezzük el őket, hogy -mind vízszintesen, mind függőlegesen- az egymástól egységnyi távolságra lévő, azaz csak egyetlen bit-ben különböző mintermeket reprezentáló négyzetek szomszédosak legyenek, vagyis így bináris kódszavuk ún. Hamming-távolsága '1'. Ennek az a nagy előnye, hogy igen könnyen észrevesszük az összevonható mintermeket illetve termeket, hiszen azok egymás mellett helyezkednek el.

Attól függően, hogy egy adott függvényt illetve mintermet hány változó határoz meg, az adott bitszám szerint nevezzük el az elkészített táblázatot (pl. kétváltozós Karnaugh tábla, háromváltozós Karnaugh tábla stb.)

Példák:

- *kétváltozós Karnaugh tábla*

Két változó (A, B) esetén 4 lehetséges mintermünk van. Ezeket egy 2×2 -es négyzeten helyezzük el (2.1.1.4.a ábra). Az egyes négyzetek jobb alsó sarkában megjelöltük az adott binárisan kódolt mintermhez tartozó decimális számértéket is.

Megjegyzés: általános elvként alkalmazzuk a felsorolási sorrend szerinti helyiérték értelmezést, vagyis esetünkben ' AB ' kétváltozós függvényben ' A ' képezi a legmagasabb

helyiértékű bitet (MSB, =**M**ost **S**ignificant **B**it). A további példákban is ezt az értelmezést használjuk.

		A	
		0	1
B	0		
	1		

		0	2
		1	3

2.1.1.4.a ábra[1]

Kétváltozós (A, B) Karnaugh tábla

- *háromváltozós Karnaugh tábla*

Három változó esetén 8 lehetséges mintermünk van. Ezeket egy 4×2 -es téglalapon helyezzük el, megfelelő peremezéssel. A peremezés azt jelenti, hogy a változókat két csoportra osztjuk, egy kéttagú és egy egytagú csoportra. Vízszintesen elrendezzük az első csoport lehetséges kombinációit (0 0, 0 1, 1 1, 1 0), függőlegesen pedig a második

egyváltozós csoport két lehetséges értékét, (0, 1). Figyeljünk fel arra, hogy a mintermek geometriai szomszédossága csak úgy biztosítható, ha a csoportok kombinációi is egységnyi Hamming-távolságra vannak egymástól. Ezért a különleges és szokatlan sorrend, azaz (0 1) után az (1 1) következik!

Nézzük az 2.1.1.4.b ábrát! Ezzel a peremezéssel a négyzetek megcímezhetők a változókhoz rendelt bináris vektorokkal, ugyanakkor a négyzetek (cellák) jelölhetők is a bináris vektornak megfelelő decimális számjeggyel. Például a felső négyzetsor harmadik négyzete az (1 1 0) bináris vektorral címezhető, és a 6-os decimális értékkel jelölhető. A minterm algebrai alakját az ismert módon olvassuk ki:

$$A B \bar{C}$$

A Karnaugh táblákhoz tartozó bináris változók logikai értékei –az adott sorokban és oszlopokban- másfajta jelöléssel is megadhatóak. Erre mutat példát a 2.1.1.4.c ábra szerinti táblázatforma.

		A			
		B			
		0	0	1	1
C	0			$A B \bar{C}$	
	1				
		0	2	6	4
		1	3	7	5

2.1.1.4.b ábra

Háromváltozós Karnaugh tábla

		A			
		B			
		0	0	1	1
C	0			$A B \bar{C}$	
	1				
		0	2	6	4
		1	3	7	5

2.1.1.4.c ábra

Háromváltozós Karnaugh táblázat más logikai érték jelöléssel

A táblázatok szerkesztésénél figyeljünk arra is, hogy nem minden logikai szomszédosság jelenik meg geometriai szomszédossággként. Beláthatjuk, hogy a (0 0 0) szomszédos az (1 0 0) mintermmel, a (0 0 1) pedig az (1 0 1) mintermmel, bár nincsenek egymás mellett. Ezt szem előtt kell tartanunk a használatnál, de beláthatjuk, hogy egy henger palástjára csavarva a táblát, a geometriai és logikai szomszédosság együttállása tökéletes lenne.

- *négyszéletes Karnaugh tábla*

A négyszéletes Karnaugh tábla összeállítása és peremezése elvi újdonságot nem jelent, de a széleken elhelyezkedő mintermek közötti szomszédosságokat –az előzőekben említett okból kifolyóan- érdemes megvizsgálni (2.1.1.4.d ábra).

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

2.1.1.4.d ábra[1]

Négyszéletes Karnaugh tábla

Az egyszerűsítés elve, azaz a lehető legnagyobb összevont termék kiolvasása a táblázatból

Példa:

egy négyszéletes Karnaugh táblában az 5-ös és 13-as (decimális számértékű) mintermekhez '1'-et rendel egy teljesen határozott logikai függvény (2.1.1.4.e ábra). Az összevonhatóság elvét követve az egyesített (összevont) termék itt egy kétnégyzetes téglalap, amelynek a címéből az 'A' változó értéke már hiányzik, hiszen az 'A' az a változó, amely a VAGY művelettel kiesik.

Így az összevont cím algebrai alakja :

$$B \overline{C} D$$

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

2.1.1.4.e ábra[1]

Összevont term létrehozása

Megjegyzés: a táblázatban felesleges a '0'-ás minterm értékek jelölése, így az - a könnyebb átláthatóságot szem előtt tartva - elhagyható.

Tegyük most fel, hogy az eddigi mintermeken kívül szerepel a függvényben a 7-es és a 15-ös is (2.1.1.4.f ábra). Ez utóbbiakat egymással összevonva felismerjük, hogy az így kialakult két mintermes két term ugyancsak szomszédos, és a C változó is kiejthető. Ezzel egy geometriailag még nagyobb, négy mintermes term adódott az összevonással, amely már csak két változós. Kiolvassa:

$$B D$$

		<i>A</i>		0	0	1	1
		<i>B</i>		0	1	1	0
<i>C</i>	<i>D</i>						
0	0						
		0	4	12	8		
0	1		1	1			
		1	5	13	9		
1	1		1	1			
		3	7	15	11		
1	0						
		2	6	14	10		

2.1.1.4.f ábra[1]

Két szomszédos term összevonásával képzett kétváltozós term

2.1.2 Teljesen és nem teljesen határozott függvények egyszerűsítése Karnaugh táblán

2.1.2.1 Teljesen specifikált függvények egyszerűsítése a Karnaugh tábla segítségével

A tervezés lépései

1. lépés: a szükséges méretű(változószámú) Karnaugh tábla felvétele
2. lépés: a függvény '1' értékeihez tartozó mintermek bejelölése
3. lépés: az összes prímmimplikáns-terület kijelölése összevonással
4. lépés: az egyszerűsített függvény felírása a rendelkezésre álló prímmimplikánsok közül történő válogatással

Példa:

egyszerűsítsük Karnaugh táblán a korábbi példában szereplő, teljesen határozott függvényünket!

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

Az 1. a 2. és a 3. lépést illusztráljuk az 2.1.2.1.a ábrán. Az '1'-es mintermek bejelölése után megkezdődhet a párosával történő összevonás. Mivel tagjaik szomszédosak, összevonhatók a következő párok:

$$(0\ 1\ 0, 0\ 1\ 1), (0\ 1\ 0, 1\ 1\ 0), (1\ 1\ 0, 1\ 0\ 0).$$

Ezután azt vizsgáljuk, hogy a három összevont kettes csoportok között vannak-e szomszédos, nagyobbakká összevonhatók. Szembetűnő, hogy nincsenek ilyenek, tehát a három kétváltozós term mindegyike prímisszimplikáns.

A 4. lépés ugyancsak a Karnaugh táblán végezhető el a legegyszerűbben. Képzeljük el, hogy a prímisszimplikánsok négyszögei lemezek, amelyek felemelhetők a tábláról. Ha ezeket sorra felemeljük, és azt találjuk, hogy az felemelt term által fedett valamennyi minterm a többi term által fedve marad, akkor a felemelt term felesleges, eltávolítható. Ezzel szemben, ha a felemelés következtében valamelyik minterm fedetlen marad, a prímisszimplikáns elhagyhatatlan.

		<i>A</i>		0	0	1	1
		<i>B</i>		0	1	1	0
<i>C</i>	0			1	1	1	
	1		1				

2.1.2.1.a ábra[1]

Lehetséges prímisszimplikánsok

Az egyszerűsítéshez szóba jöhető prímisszimplikánsok:

$$\overline{A}B, B\overline{C}, A\overline{C}$$

A teljes lefedettség szükséges és elégséges elvét követve látható, hogy

$$B\overline{C}$$

elhagyható, felesleges term.

2.1.2.2 Nem teljesen specifikált függvények egyszerűsítése a Karnaugh táblán

Ha a logikai függvény nem teljesen határozott, akkor legalább egy olyan bemeneti kombináció, azaz minterm van, amelyhez rendelt függvény érték számunkra közömbös. Ilyenkor különböző szimbólumokkal jelöljük be a Karnaugh-táblába az '1'-es és közömbös (don't care) mintermeket. Ez utóbbiakat célszerű a már ismert kis vízszintes vonalkával (–) jelölni. A közömbös mintermekkel szabadon bánhatunk. Ha előnyös az egyszerűsítés szempontjából, akkor összevonjuk őket az '1'-es mintermekkel, ha nem, akkor '0'-ás mintermeknek tekintjük őket.

Példa:

adott egy négyváltozós (A,B,C,D) , 'közömbös' mintermekkel is rendelkező logikai függvény Karnaugh táblás alakja (2.1.2.2.a ábra):

		A		0	0	1	1
		B		0	1	1	0
CD							
00							1
01			1		1		–
11		–		–	–		–
10							

2.1.2.2.a ábra[1]

Egy nem teljesen specifikált, négyváltozós logikai függvény Karnaugh táblás alakja, a közömbös bejegyzések közötti válogatással előállított prímimplikánsok megadásával

A prímiimplikánsok között egy „négymintermes”, tehát kétváltozós, és a közömbös bejegyzéssel rendelkező lehetséges prímiimplikánsok közötti válogatással előállított két ’kétmintermes’, vagyis háromváltozós term szerepel. Kiolvassva:

$$BD, \quad A\bar{C}D, \quad A\bar{B}\bar{C}$$

Ezek közül a második elhagyása nem változtat a teljes lefedésen.

(Megjegyzés: a függvényalakban a teljes lefedettséget biztosító prímiimplikánsok mellett felírt felesleges, ún. redundáns prímiimplikánsok alkalmazása a gyakorlati, kapusintű megvalósítás során bizonyos esetekben szükséges lehet, melynek pontos okaival a későbbiekben részletesen foglalkozni fogunk.)

A tervezés lépései

1. lépés: a szükséges méretű (változószámú) Karnaugh tábla felvétele
2. lépés: a függvény '1' és 'közömbös' értékeihez tartozó mintermek bejelölése
3. lépés: a közömbös bejegyzéssel is rendelkezhető lehetséges prímiimplikánsok kijelölése
4. lépés: az egyszerűsített függvény felírása a rendelkezésre álló prímiimplikánsok közül történő válogatással

Felhasznált irodalom:

[1] Dr. Keresztes Péter: Digitális hálózatok (HEFOP elektronikus Jegyzet)