DIGITÁLIS TECHNIKA I

Dr. Lovassy Rita
Dr. Pődör Bálint

Óbudai Egyetem KVK
Mikroelektronikai és Technológia Intézet

7. ELŐADÁS

1

Arató Péter: Logikai rendszerek tervezése, Tankönyvkiadó, Budapest, 1990, Műegyetemi Kiadó 2004, 55013 műegyetemi jegyzet

Zsom Gyula: Digitális technika I és II, Műszaki Könyvkiadó, Budapest, 2000, (KVK 49-273/I és II)

Rőmer Mária: Digitális rendszerek áramkörei, Műszaki Könyvkiadó, Budapest, 1989, (KVK 49-223)

Rőmer Mária: Digitális technika példatár, KKMF 1105, Budapest 1999

Az előadás ezen könyvek megfelelő fejezetein alapszik.

A SZÁMJEGYES MINIMALIZÁLÁS ALAPJAI
Quine-McCluskey módszer

Algoritimizálható, programozható.

Logikai függvényegyszerűsítéshez a Karnaugh-táblák használata korlátozott:
– <5 változós függvények
– Egyszerre egyetlen kimeneti függvény
– Szubjektív megközelítés, különböző eredmények

Mintermek: alsó indexek egyértelműen megadják. Csupán ezek ismeretén alapuló minimalizálási eljárás: a végrehajthatóság nem függ a változók számától.

3

MINTERMEK SZOMSZÉDOSSÁGA

A minimalizálás alapja a szomszédos mintermek megkeresése, egyszerűsítése, míg el nem jutunk a prímimplikánsokig.

Két minterm szomszédosságának szükséges és elégséges feltétele három állítással adható meg, melyeknek egyszerre kell, hogy teljesülniük.

Lényeges, hogy e feltételek megfogalmazhatók kizárólag a mintermek alsó indexei értékeire alapozva.

4

KÉT MINTERM SZOMSZÉDOSSÁGÁNAK FELTÉTELE (1)

Két minterm szomszédos, ha decimális indexeik különbsége 2 egész számú hatványa.

6	0110	$\bar{A}\,B\,C\,\bar{D} + \bar{A}\,\bar{B}\,C\,\bar{D} \rightarrow \bar{A}\,C\,\bar{D}$
2	0010	
4		

Ez szükséges de nem elégséges feltétel, mivel pl. a 2 és 4 indexű mintermekre is teljesül, de ezek nem szomszédosak.

5

KÉT MINTERM SZOMSZÉDOSSÁGÁNAK FELTÉTELE (2)

Két minterm szomszédos, ha bináris súlyaik (1-esek száma) különbsége 1.

6	0110 (2)	$\bar{A}\,B\,C\,\bar{D} + \bar{A}\,\bar{B}\,C\,\bar{D} \rightarrow \bar{A}\,C\,\bar{D}$
2	0010 (1)	
4	(1)	

Egyikük egyel és csakis egyel több 1-est tartalmaz bináris formájában. Ez is szükséges de nem elégséges feltétel, mivel pl. a 2 és 4 indexű mintermekre, bár a decimális különbség 2 hatványa, éppen ez a feltétel mely nem teljesül.

6

KÉT MINTERM SZOMSZÉDOSSÁGÁNAK FELTÉTELE (3)

A két minterm szomszédos, ha a nagyobb bináris súlyú mintermnek a decimális indexe is nagyobb a másikénál.

6	0110 (2)	$\bar{A}\,B\,C\,\bar{D} + \bar{A}\,\bar{B}\,C\,\bar{D} \rightarrow \bar{A}\,C\,\bar{D}$
2	0010 (1)	
4	(1)	

$6 > 2$ és $2 > 1$

Ez is szükséges de önmagában nem elégséges feltétel, mivel pl. a 7 és 9 indexű mintermekre, melyekre az első két feltétel áll, éppen ez nem teljesül, persze ezek nem szomszédosak.

7

QUINE-MCCLUSKEY ALGORITMUS

Bizonyítható azonban, hogy ezen három feltétel egyidejű teljesülése már nemcsak szükséges hanem elégséges feltétele a két term szomszédosságának.

Ezen alapul a Quine-McCluskey algoritmus.

8

QUINE-MCCLUSKEY ALGORITMUS

A számjegyes minimalizálás Quine-McCluskey féle algoritmusa ezen három feltétel alapján, kizárólag a mintermek indexeit vizsgálva válogatja párba a mintermeket, majd egyszerűsítés után a folyamatot addig ismétli míg el nem jut a prímimplikánsokig.

Az algoritmus az összes prímimplikánst eredményezi így a második lépés az, hogy ki kell választani közülük a lényeges prímimplikánsokat.

Az algoritmus gyakorlati alkalmazását egy példa mutatja be.

9

QUINE-MCCLUSKEY MINIMALIZÁLÁS

Minimalizálandó függvény:

$$f(A,B,C,D) = \Sigma m(0,2,3,5,7,8,10,13,15)$$

$$|\text{minterms}| = 0 \Rightarrow 0$$
$$= 1 \Rightarrow 2,8$$
$$= 2 \Rightarrow 3,5,10$$
$$= 3 \Rightarrow 7,13$$
$$= 4 \Rightarrow 15$$

A mintermeket az indexeik bináris vagy Hamming súlya szerint csoportosítjuk

http://cpe.gmu.edu/courses/ece331/lectures/331_8/index.htm

10

MINTERM TÁBLA

Súly	Minterm		
0	0		
1	2 8		
2	3 5 10		
3	7 13		
4	15		

11

SZOMSZÉDOK MEGKERESÉSE…

Két minterm szomszédos, ha decimális indexeik különbsége 2 egész számú hatványa.

Súly	Minterm	Párok	
0	✓0	0,2 (2) 0,8 (8)	
1	✓2 ✓8		
2	3 5 10		
3	7 13		
4	15		

Összevonás szomszédos csoportok között, ha az indexek különbsége 1, 2, 4, 8, stb. Felhasznált termek megjelölendők. Egy term több párban is szerepelhet.

12

AZ ÖSSZES SZOMSZÉDPÁR

Súly	Minterm	Pár	
0	✓0	0,2 (2) 0,8 (8)	
1	✓2 ✓8	2,3 (1) 2,10 (8) 8,10 (2)	
2	✓3 ✓5 ✓10	3,7 (4) 5,7 (2) 5,13 (8)	
3	✓7 ✓13	7,15 (8) 13,15 (2)	
4	✓15		

Ezután a párokat kell párosítani: 4-es csoportok

13

NÉGYES CSOPORTOK

Két minterm szomszédos, ha bináris súlyaik (1-esek száma) különbsége 1.

Kék csillag * : prímimplikáns

Ezek a tagok nem lettek összevonva

Súly		Minterm	Pár	Négyes
0		✓0	✓0,2 (2) ✓0,8 (8)	0,2,8,10 (2,8)
1	*	✓2 ✓8	2,3 (1) ✓2,10 (8) ✓8,10 (2)	
2	*	✓3 ✓5 ✓10	3,7 (4) ✓5,7 (2) ✓5,13 (8)	5,7,13,15 (2,8)
3		✓7 ✓13	✓7,15 (8) ✓13,15 (2)	
4		✓15		

14

PRÍMIMPLIKÁNS TÁBLA

Prím-implikánsok	Mintermek 0 2 3 5 7 8 10 13 15
0,2,8,10 (2,8) 5,7,13,15 (2,8)	
2,3 (1) 3,7 (4)	

15

PRÍMIMPLIKÁNS TÁBLA

Prím-implikánsok	Mintermek 0 2 3 5 7 8 10 13 15
0,2,8,10 (2,8) 5,7,13,15 (2,8)	X X X X X X X X
2,3 (1) 3,7 (4)	X X X X

Az m0 minterm csak egy sorban fordul elő, valamint, m8 és m10 is, ezért m(0,2,8,10) lényeges prímimplikáns. Ez lefedi majd az m2 mintermet is.

16

PRÍMIMPLIKÁNS TÁBLA

Prím-implikánsok	Mintermek 0 2 3 5 7 8 10 13 15
★0,2,8,10 (2,8) ★5,7,13,15 (2,8)	X X X X X X X X
2,3 (1) 3,7 (4)	X X X X

17

MINTERMEK LEFEDÉSE

Prim-implikánsok	√ √ √ √ √ √ √ √ 0 2 3 5 7 8 10 13 15
★0,2,8,10 (2,8) ★5,7,13,15 (2,8)	X X X X X X X X
2,3 (1) 3,7 (4)	X X X X

A két négyes prímimplikáns az m3 kivételével már lefedi az összes mintermet.

18

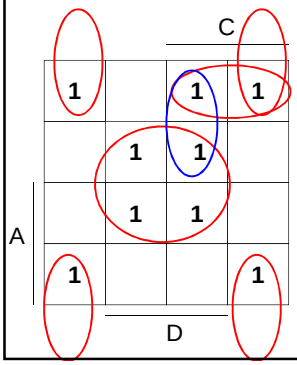
MINIMALIZÁLÁS EREDMÉNYE

$$f(A,B,C,D) = 0,2,8,10(2,8) + 5,7,13,15(2,8) + 2,3(1)$$
$$= X0X0 + X1X1 + 001X$$
$$= \overline{B}\overline{D} + BD + \overline{A}\overline{B}C$$

0	0000	5	0101	2	0010
2	0010	7	0111	3	0011
8	1000	13	1101		
10	1010	15	1111		
	X0X0		X1X1		001X

19

MEGOLDÁS A KARNAUGH TÁBLÁN



Lényeges prímimplikánsok

Redundás (nem lényeges) prímimplikáns)

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

QUINE-MCCLUSKEY ALGORITMUS PROGRAM

www.seattlerobotics.org/encoder/200106/qmccmin.htm

Példa: 64 változós függvény 64 mintermet tartalmazó alakjának minimalizálása

21

Kombinációs hálózatok megvalósítása memóriaelemek felhasználásával

22

MEMÓRIAELEMEK TULAJDONSÁGAI

Állandó tartalmú memóriák:

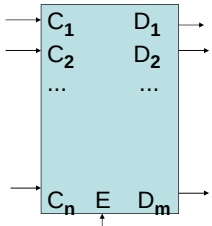
Read Only Memory (ROM),

tulajdonságaik alapján alkalmasak kombinációs hálózatok megvalósítására.

A memóriaelemben tárolt adat egy bináris kombináció (D_1, D_2, ... D_m) , mely a cím megadásával, mely szintén bináris kombináció (C_1, C_2, ... C_n), válik hozzáférhetővé.

23

MEMÓRIAELEM ELVI VÁZLATA



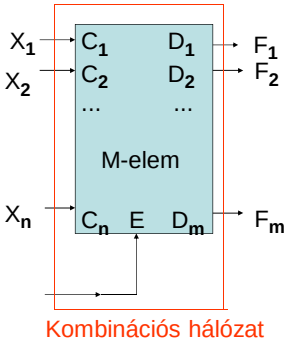
C – címbemenet
D – adatkimenet
E - (enable, engedélyező)

A C bemenetre érkező n-bites kombináció hatására a D kimeneten megjelenik a megfelelő cellában tárolt m-bites kombináció.

Az E (enable, engedélyező) bemenetre adott jel letiltja vagy engedélyezi az adatkimenetet. Ennek révén, és a memória-elem áramköri kialakítása miatt több memóriaelem kimenetei huzalozott VAGY kapcsolat szerint közösíthetők.

24

MEMÓRIAELEM MINT KOMBINÁCIÓS HÁLÓZAT

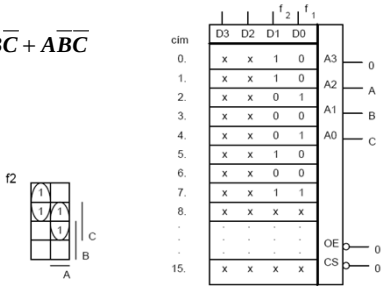


A memóriaelem egy n-változós és m kimeneti függvénnyel bíró kombinációs hálózatot valósít meg.

Az adatbeírás közvetlenül az igazságtáblázat alapján végezhető el, nem szükséges a minimalizálás.

ROM MINT UNIVERZÁLIS KOMBINÁCIÓS HÁLÓZAT

$$f_1 = ABC + \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C}$$



$$f_2 = \overline{A}\overline{B} + AC = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}C + ABC$$

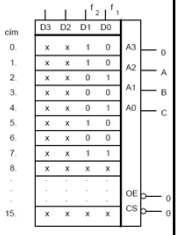
Mintermek: f_1: 2, 4, 7, illetve f_2: 0, 1, 5, 7

ROM MINT KOMBINÁCIÓS HÁLÓZAT

A két 3-változós függvényhez egy 8x2 bites ROM elegendő, ilyen nincs forgalomban, 16x4 bitest alkalmazunk.

A2, A1, A0 cím – A, B, C változók
D0, D1 kimenet – f1, f2 függvény

Igazságtábla előállítása és beprogramozása:
A3 címbemenet fixen 0-ra kötve (csak a ROM alsó 8 szavát használjuk, a többi terület közömbös)
D3, D2 – közömbös
CS, OE fixen aktív szintre kötve, folyamatos engedélyezés



ROM ALKALMAZÁSOK: KÓDÁTALAKÍTÁS

A ROM-ok egyik legfontosabb felhasználási területe a kombinációs logikában a kódátalakítás.

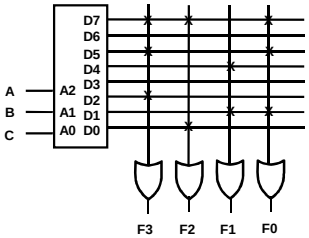
n-bites kód → m-bites kód → szükséges memóriakapacitás m x 2^n.

8-bites bináris → 8-bites Gray kód: 8x256 ROM.

READ ONLY MEMORY: PÉLDA

Példa: N = 3 bemenet, M= 4 kimenet
= Fix "AND" kapuszint + egy 3/8 dekóder,
A 8 kimenet valósítja meg a mintermeket.

A programozható illetve megcímezhető "OR" kapuszinteken a betárolt „1”-esek felelnek meg az egyes, az OR kapuk bemeneteire kötött AND kapuknak.



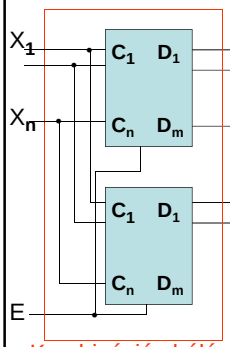
- F3 = D7 + D5 + D2
- F2 = D7 + D0
- F1 = D4 + D1
- F0 = D7 + D5 + D1

DINAMIKUS VISELKEDÉS, HAZÁRDOK

M-elemekre a statikus és dinamikus hazárd nem értelmezhető a kapuhálózatokhoz hasonló módon, a hazárdmentesítési eljárások sem alkalmazhatók.

Lényeges az, hogy a címváltozás időpontjától számított ciklusidő múlva az adat rendelkezésre áll a kimeneten. A ciklusidőn belüli tranziens kimeneti változások zavaró hatását szinkronizációval vagy vezérléssel kell kiküszöbölni. Erre jól alkalmazható az Enable bemenet.

M-ELEMEK ÖSSZEKAPCSOLÁSA



Két m kimenetű m-elem cím- és engedélyező-bemeneteinek megfelelő összekapcsolásával m+k kimenet valósítható meg (az ábrán k = 2).

A bemeneti jelek mindkét m-elem címbemeneteire eljutnak, a kimeneten a pótlólagos két bitet a második elem adja.

31

Kombinációs hálózatok megvalósítása programozható logikai elemek (PLD) felhasználásával

32

PROGRAMOZHATÓ LOGIKAI ESZKÖZÖK

PLD: logikai hálózatok realizálására szolgáló általános célú IC chip.

Különféle logikai elemeket tartalmaz, melyek közötti összekötetés többféleképen is kialakítható.

Egy „fekete doboz”-nak tekinthető mely logikai elemeket (kapukat) és „programozható” kapcsolókat tartalmaz.

Bármilyen logikai hálózat realizálható, az adott eszközön lévő elemek fajtája és szám által korlátozottan.

33

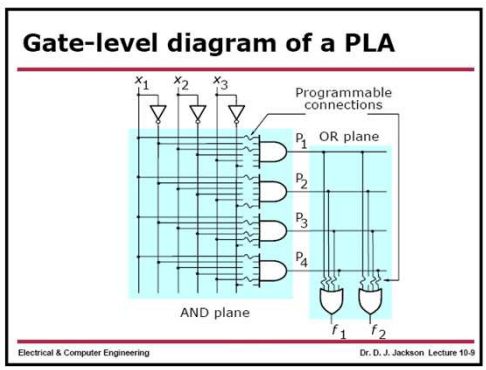
PROGRAMOZHATÓ LOGIKAI ESZKÖZÖK: PLD

Programozható logikai eszköz- Programmable Logic Device (PLD) – egyik leggyakoribb formája a programozható logikai mátrix (Programmable Logic Array, PLA): adott számú ÉS kapu, ill. VAGY kapu és adott számú INVERTER.

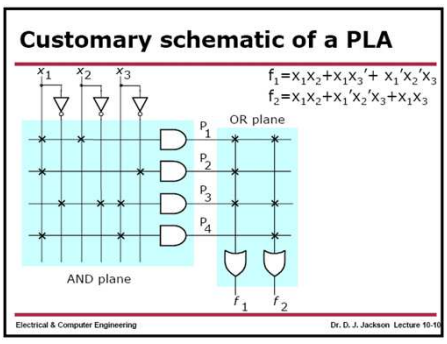
A belső összekötetéseket a kétszintű több kimenetű kombinációs hálózat logikai rajza alapján hozzák létre. Az elvi logikai rajz megadja, hogy az egyes változók a PLA mely bemeneteire jussanak.

Az elvi logikai rajz létrehozásánál az ismert minimalizálási és hazárdmentesítő eljárások alkalmazhatók.

34



35



36

PLA ELEMEK TULAJDONSÁGAI ÉS ALKALMAZÁSA

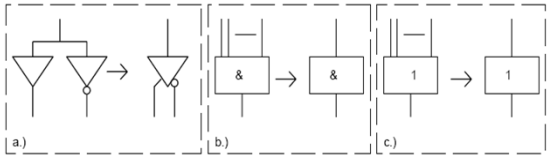
A PLA elem alábbi jellemzői megszabják, hogy egy adott kombinációs hálózat megvalósításához elég egy PLA elem, vagy nem:

- a PLA bemeneteinek száma;
- a PLA elem kimeneteinek száma;
- a belső ÉS kapuk száma.

A belső ÉS kapuknak általában annyi bemenete van ahány a PLA elem bemeneteinek száma.
A belső VAGY kapuk általában annyi bemenetűek, amennyi a belső ÉS kapuk száma.
Van PLA bemeneti INVERTER nélkül is, ekkor a ponált és negált bemeneti jelek mindegyike két PLA bemenetet foglal le.

37

PLA: SZOKÁSOS JELÖLÉSEK

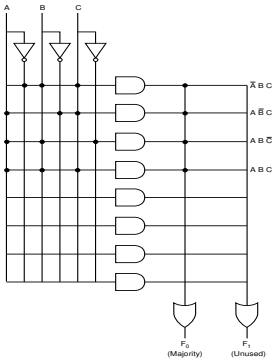


38

PÉLDA: A MAJORITÁS FÜGGVÉNY PLA REALIZÁLÁSA

3-változós majoritás függvény (az 1-esek száma több mint a 0-áké)

M = Σ(3,5,6,7)



39

PLA ELEMEK ÖSSZEKAPCSOLÁSA (1)

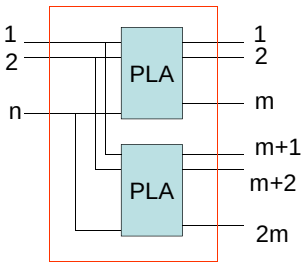
Ha a megvalósítandó kombinációs hálózat elvi logikai rajza nem minimalizálható tovább és egyetlen PLA elemmel nem valósítható meg, akkor több PLA elemet kell megfelelő módon összekapcsolni.

A PLA elemek összekapcsolási módja attól függ, hogy melyik PLA jellemző a korlátozó tényező.

40

PLA ELEMEK ÖSSZEKAPCSOLÁSA (2)

Ha csak a PLA elem kimeneteinek száma kevés, akkor pl. két PLA elem bemeneteit kapcsoljuk össze páronként.



41

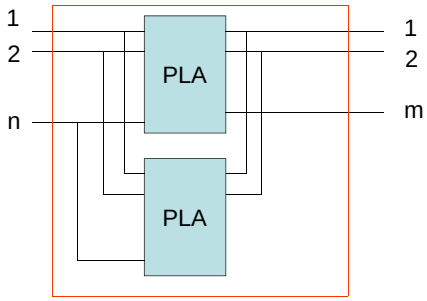
PLA ELEMEK ÖSSZEKAPCSOLÁSA (3)

Ha csak a belső ÉS kapuk száma nem elegendő akkor a bemenetek páronkénti összekapcsolása mellett a kimenetek is páronként huzalozott VAGY-ként összekapcsolandók.

Ha áramköri okokból a huzalozott VAGY nem használható, akkor külső VAGY kapukkal vagy további PLA elemmel kell a szükséges VAGY kapcsolatot létrehozni.

42

PLA-K KAPCSOLÁSA NEM ELEGENDŐ BELSŐ „ÉS” KAPUK ESETÉN



43

PLA ELEMEK ÖSSZEKAPCSOLÁSA (4)

Az eddig nem vizsgált esetek és azok az esetek, mikor egyidejűleg több korlátozó tényező hatását kell kiküszöbölni, lényegében hasonlóan kezelendők.

Erre további példák és útmutatás Arató Logikai rendszerek tervezése könyvében találhatók.

44

PLA PROGRAMOZÁSA, FPLA

A PLA programozása lehet maszkprogramozott, az összekötetéseket a gyártás során a fémezési maszk megfelelő kialakításával rögzítik, ez később már nem változtatható.

A felhasználó által programozható típus a Field Programable Logic Array (FPLA). A programozás a beépített ”biztosítékok” megfelelő árammal való ”kiégetésével” történik.

45

PROGRAMABLE ARRAY LOGIC

A Programable Array Logic (PAL) áramkörnek nevezett FPLA elem egyszerűbb mint maga a PLA, mivel csak az ÉS kapui programozhatók, a VAGY kapuk bekötése a gyártó által rögzített. Nevezik programozható logikai mátrix-nak is.

Az ÉS mátrixban Schottky diódák (fém-félvezető diódák) vannak, a diódák túlfeszültség hatására elégethető vezetőréteggel csatlakoznak az összekötő vezetékekhez.

46

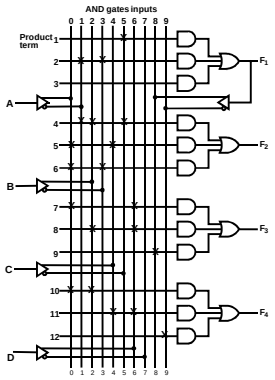
Programmable Array Logic Example

$F1 = \overline{A}\,\overline{B} + \overline{C}$

$F2 = \overline{A}\,\overline{BC} + AC + \overline{AB}$

$F3 = F3 = AD + \overline{BD} + F1$

$F4 = AB + CD + \overline{F1}$



47

PLA: HAZÁRDOK KEZELÉSE

Egyetlen PLA elemmel felépített kombinációs hálózatnál a programozásnál a kétszintű elvi kapcsolási rajzból kell kiindulni, amelynek hazárdmentesítését az ismert eljárásokkal lehet és kell elvégezni.

Több PLA elemből felépített hálózatoknál további ellenőrzések válhatnak szükségessé.

48

M-ELEMET ÉS PLA-T HASZNÁLÓ MEGOLDÁSOK ÖSSZEHASONLÍTÁSA

Működési sebesség:

PLA-val nagyobb működési sebesség érhető el.

Okai: a PLA lényegében kétszintű kombinációs hálózat, továbbá technológiai okokból a PLA elem eleve gyorsabb működésű mint a memóriaelem.

49

M-ELEMET ÉS PLA-T HASZNÁLÓ MEGOLDÁSOK ÖSSZEHASONLÍTÁSA

Hazárdmentesség:

PLA esetén a hazárdmentessége a tervezés során az ismert eljárások alkalmazásával biztosítható.

A memórialemek ciklusidőn belüli viselkedése nem definiált!
Megoldás: vezérlés, illetve szinkronizálás.

50

M-ELEMET ÉS PLA-T HASZNÁLÓ MEGOLDÁSOK ÖSSZEHASONLÍTÁSA

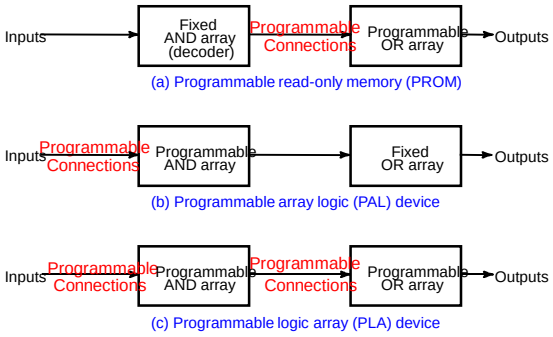
Tervezési folyamat:

A tervezés menete a memóriaelemek esetén egyszerűbb, mivel közvetlenül az igazságtáblázaton (diszjunktív kanonikus alak) alapul.

A PLA esetén a programozáshoz létre kell hozni az egyszerűsített hazárdmentes kétszintű elvi logikai rajzot.

51

ROM, PAL ÉS PLA ERENDEZÉSE



52

M-ELEM, PLA, PAL ÖSSZEHASONLÍTÁSA

	ÉS kapu	VAGY kapu
Memória elem	fix	program
PLA	program	program
PAL (FPLA)	program	fix

53

KOMBINÁCIÓS HÁLÓZATOK TERVEZÉSE: ÁLTALÁNOS ELVEK

1. A megoldandó feladat alapos áttekintése és megértése
Mit kell az áramkörnek megoldani?
Mik a bementek (adatok vagy logikai változók, vezérlés) és a kimenetek
Funkcionális vagy blokk diagram elkészítése

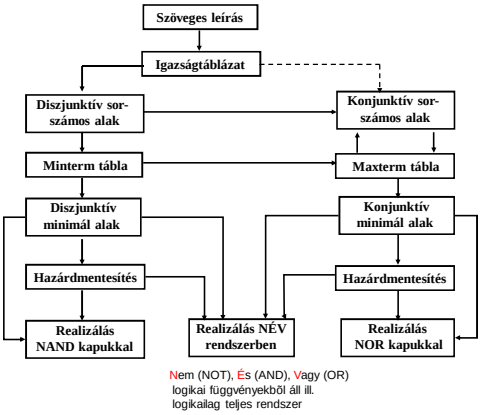
2. A feladat valamely alkalmas logikai tervezési reprezentációban való megfogalmazása
Legtöbbször igazságtáblázat vagy idődiagram formájában
Esetleg a szimbólikus be- és kimeneti változókat kódolni is kell

54

KOMBINÁCIÓS HÁLÓZATOK TERVEZÉSE: ÁLTALÁNOS ELVEK

3. Megvalósítási technológia, technika és hardware megválasztása
ROM, PAL, PLA
Multiplexer, dekóder és VAGY kapu
Diszkrét kapuk: kétszintű ÉS-VAGY, illetve VAGY-ÉS, többszintű logika, XOR logika (KIZÁRÓ-VAGY), stb.

4. A kiválasztott rendszerhez tartozó tervezési/realizálási eljárás alkalmazása
Karnaugh táblázat vagy numerikus minimalizálás két- vagy többszintű hálózatoknál (diszkrét kapuk vagy PLD), hazárdmentesítés
Tervező rendszerek és hardware leíró nyelvek (pl. Verilog) használata nagyobb és komplex rendszereknél



Kapuáramkörökön alapuló tervezés folyamatábrája