

8. tétel

- Minden olyan állítást, amely 1 v. 0 logikai értékekkel nem igazolható.
- Egy kijelentés logikai értéke a feltétel esetén szigorúan vagy az egyik, vagy a másik, kölcsönösen kizárólag egymást.
- az i v. k tulajdonságot logikai értékkel nem igazolható.
- Továbbá: 1: igaz, 0: hamis
- Ha az $E = \{0, 1\}$ halmazon műveletek definiáltak, \Rightarrow algebrai struktúrát kapunk. A művelet nem vezet ki a halmazból.
- Ez az algebrai kijelentések által felvehető értékek algebraja, azaz: logikai algebra v. bináris algebra.
- 1854: Bevezette a gondolkodás törvényeit (George Boole) \Rightarrow bináris alg. alapjai

- 3 alapvető művelet definiálva

konjunkció: logikai AND \wedge (AND)

diszjunkció: logikai VAGY \vee (OR)

negáció: tagadás \neg (NOT)

A logikai AND művelettel eredménye $\Leftrightarrow 1$, ha mindkét komponens igaz.



A logikai VAGY művelettel eredménye $\Leftrightarrow 1$, ha valamely komponens igaz.



A logikai tagadás művelettel eredménye a logikai érték ellentettje.



Tulajdonságok:

asszociativitás: $x_1 \cdot x_2 = x_2 \cdot x_1$; $x_1 + x_2 = x_2 + x_1$

kommutativitás \rightarrow $x_1 (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3$; $x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$

distributivitás: $x_1 (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$; $x_1 + (x_2 \cdot x_3) = (x_1 + x_2) \cdot (x_1 + x_3)$

adnórció: $x_1 \cdot (x_1 + x_2) = x_1$; $x_1 + (x_1 \cdot x_2) = x_1$

tautológia: $x_1 \cdot x_1 = x_1$; $x_1 + x_1 = x_1$

negáció : $x_1 \cdot \neg x_1 = 0$; $x_1 + \neg x_1 = 1$

Értős negáció: $\neg \neg x_1 = x_1$

De Morgan azon.: $\neg(x_1 + x_2) = \neg x_1 \cdot \neg x_2$; $\neg(x_1 \cdot x_2) = \neg x_1 + \neg x_2$

művelet 1-gyel és 0-val: $x_1 \cdot 1 = x_1$; $x_1 + 1 = 1$; $\neg 1 = 0$; $\neg 0 = 1$
 $x_1 \cdot 0 = 0$; $x_1 + 0 = x_1$

- Felcsinálva az egyenletpárokat, látható a DUALITÁS törvénye. Ha

↑ azonosítjuk felcsináljuk a 0-ot a 1-gyel és a 1-et az 0-gal ⇒
mindkét azonosítást lépünk.

9. tétel

- Ha egy x változó 1 és 0 értéket vehet fel, akkor mondjuk,
a x BINÁRIS VÁLTOZÓ v. LOGIKAI VÁLTOZÓ.

- Ha egy y változó értéke az x logikai változó értékeitől
függ ⇒ x és x megfelelő értékpárai egy LOGIKAI FÜGGŐ-t hat. meg.

jel: $y = f(x)$

- Ha y logikai értéke az x_1, x_2, \dots, x_n logikai változói értékeitől
függ, akkor N-VÁLTOZÓS LOGIKAI FÜGGŐ-ről beszélünk.

jel: $y = f(x_1, x_2, \dots, x_n)$

- A függőre megadhatjuk táblázattal, képlettel, halmazrendelési
utasítással.

- n változónak 2^n lehetséges értékkombinációja van, és mivel
mindenképpen két különböző függvény tartozik, ezért a képezhető
logikai függvények száma 2^{2^n}

- két változó esetén 16 logikai függvény képezhető. A legfontosabbak:
AND, OR, XOR, NAND, NOR, ekvivalencia (\neg XOR)

- Itt digitális technikában a feladatokat legtöbbször f_{g} -táblázattal adjuk meg.
- Itt leghaladóbb kialakításához azonban szükséges van a logikai f_{g} -et legegyszerűbb alakjára, hiszen ez azt jelenti, le először kell a legegyszerűbb alakot felkerni.
- Itt ábrázoló kialakításánál a feladatnál:
 - felírni a logikai leghaladóbb táblázat formájában.
 - a táblázatból megalkotni a logikai f_{g} -et.
 - a f_{g} -et valamilyen módon a legegyszerűbb alakra koverni.
 - létrehozni a kívánt leghaladót.
- Itt f_{g} -et analitikus (éplettel megadott) alakját először leírjuk meg, ha a független logikai változókat végső számú logikai művelettel leírjuk össze. (konj., diszj., neg.) pl.: diszj. norm. alak, konj. norm. alak.
- Diszjunktív normál alak:
 - minden konj.-s tagban vagy 1 változó, v. 1 változó negáltja és minden diszj.-s tagban minden változó szerepel.
 - minden változó 1 konj.-ban 1x szerepel.
 - nincs két olyan diszj.-s tag, amelyben van a változók valamelyike eltérően.
- Konjunktív normál alak:
 - minden diszj.-s tag 1 változó v. 1 változó negáltja, és minden konj.-s tagban minden változó szerepel.
 - minden változó 1 diszj.-s tagban pontosan 1x szerepel.
 - nincs két olyan konj.-s tag, amelyben van a változók valamelyike eltérően.
- Diszj. norm. alak. $F(x_1, x_2) = (\neg x_1 \cdot \neg x_2) + (x_1 \cdot \neg x_2)$

x_1	x_2	OR
0	0	1*
0	1	0
1	0	1*
1	1	0

- Kétf. norm. alak: $F(x_1, x_2) = (x_1 + \neg x_2) \cdot (\neg x_1 + \neg x_2)$

x_1	x_2	OR
0	0	1
0	1	0*
1	0	1
1	1	0*

10. tétel

- A logikai fog. -eket egyenértékűen: algebrai, grafikai és numerikus eljárásokkal.
- Ha viszonylag kevés változóval van, \Rightarrow alkalmazható a Karnaugh-táblát (log. vált. szám ≤ 5). Ez egy grafikus eljárás.
- Két változóra:

$x_2 \backslash x_1$	$\neg x_1$	x_1
$\neg x_2$	$\neg x_1 \neg x_2$	$x_1 \neg x_2$
x_2	$\neg x_1 x_2$	$x_1 x_2$

Ez egy diszj. normál forma

- Egyenértékűen:

- egyenlős nélküli alakzat
- egész sorokat, oszlopokat
- több cellából álló négyzet, köglélapot
- a tábla "első" - négy "első" celláját

! ide is tartozik.

Grafikus minimalizálás (Karnaugh-táblát)

A igazságtábla átrendezésével egy másik táblát kapunk, amely - az a táblázat, azaz a szomszédos változókombinációk, vagyis MINITERMEK az "első" táblában helyileg a szomszédos cellákban vannak.

Szounédos mintamint azokat a mintamint nevezzük, amelyekben van egyetlen változó van ellentét állapotban.

- Egy tetszőleges logikai fgv-t úgy ábrázolhatunk Karnaugh táblán, hogy 1-t írunk azokra a cellákra, amelyekhez tartozó mintamint tartalmazza a fgv, míg a többi 0.

2 változós K.-tábla:

A \ B	0	1
0	w_0^2	w_1^2
1	w_2^2	w_3^2

Itt minimalizálás lényege $x + x\bar{f} = 1$ összefüggés alapján.

lépések: 1, csoportosítjuk kettesével a K-tábla szounédos, logikai 1-t tartalmazó celláit. (Ez a csoport a kettes körök)

2, ha lehet, a kettes köröket csoportosítjuk szounédosként, kialakítva a 4-es köröket.

3, folytatjuk a csoportosítást 4, 8, 16 körök kialakítására.

→ A köröknek megfelelő sorokat PRIMIMPLIKANSnak nevezzük.

- Általában több primimplikáns van, mint az összes mintamint legkisebb létszámú minit.

- Itt a primimplikáns, amely csak egyetlen mintamint fed le, lényeges primimplikánsnak nevezzük. Ezeket minden diszjunkt alor tartalmazza.

Itt egyszerűsített fgv felírását az egyes körökhez tartozó független változóösszeállítás alapján végezzük.

1, megvizsgáljuk, le melyik az a változó, amely nem azonos értékkel szerepel a körökbe foglalt összes mintamintban.

2, az ilyen változókat az algebrai alorból kihagyjuk.

A többi változó pedig assz. az értékel. sorozatban, amely a kiind. öm. min. értékben azonos.

3; A fgo. leggyenértékű aláírja, a leggyenértékű korlátok logikai összege len.

2. tétel

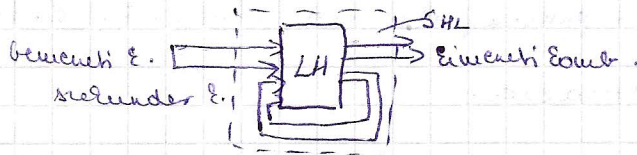
- Logikai feladat: A feladat, melynek megoldása során véges számú feltétel között kell kiválasztani és egyesek kijelöléshez egyértelműen hozzá kell rendelni minden előadás mint egy-egy következményt.

- Véges számú következmény között válogatva az olyan beosztást, amely egy logikai feladat során olyan jelölészet állít elő, amelyet a feladat megfogalmazása az éppen kijelölt feltételekhez köt, LOGIKAI HÁLÓZATNAK nevezzük.

Két típusa van: - Sorozati, referenciais log. hál. (SLH)

- komb. log. hál. (KLH)

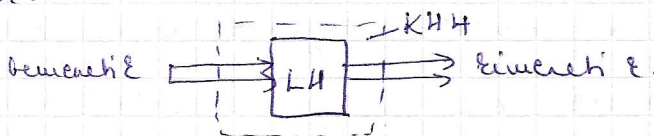
Soroz. log. hál.: Olyan logikai hálózatok, amelyek terjedési irányított tartalmazzak. Ugyanezért a bemeneti kombinációhoz más-más kimeneti kombinációk rendelnek hozzá a működés függvény kombinációi fgo.-ében.



A SLH memorizáló tulajdonsággal rendelkezik.

Komb. log. hál.:

Kimeneti kombinációja az a pillanatnyi bemeneti kombinációtól függ.



14. tétel

Összehasonlító (komparáló):

az összehasonlító olyan KLM, amelynek $2n+3$ db adatbemenete, 3 kimenete van. 2 db n bites bináris számot hasonlítanak össze, és a kimenetben a 3 lehetséges relációval megfelelő jelet artikulálja. ($=, <, >$). A 3 bemeneti relációval összehasonlító lehetőséget ad, amely négyes számok összehasonlítására szolgál.

Paritásellenőrző egység:

Olyan KLM, amelynek 2^n+2 db bemenete, 2 db kimenete van.

2^n db információkódolás létezik egyeztetve 1 paritásbittel 2^n+1

hosszú len. Ennek bennél páros és páratlan paritásról.

A páros paritásbit értéke akkor 1, ha az adatbitek közül a

logikai 1-en található értékek száma páratlan. A

páratlan paritásbit akkor logikai 0, ha az adatbitek közül

a logikai 1-en található értékek száma páratlan.

Teledata: az adatos paritásának ellenőrzése.

Memóriából való olvasás: Ha a beírási sor részesített paritásbit

és az olvasási sor részesített paritásbit

nem egyezik meg \Rightarrow az adat hibás.

ECC (Error Code Correction) hibajavító kód, 7 bit: 1 bits hibát

javítani tudja, a 2 bits hibát felismeri.

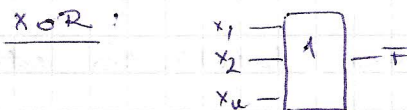
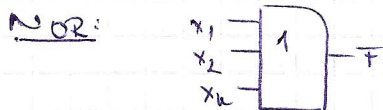
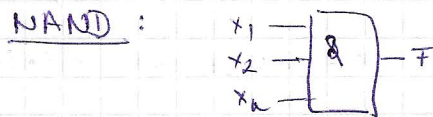
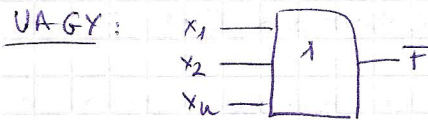
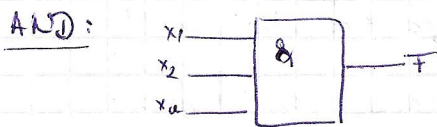
Paritáshiba van, ha az olvasott 8 bitből részesített

paritásbit és az olvasott paritásbit eltérő.

- n logikai kábelzet bemeneti és kimeneti pontjain 2^n különböző jél
lehetőség: 0, 1. Ezek a logikai értékek. Közöttük értelmezve vannak
a fő műveletek:
 - logikai AND (és)
 - logikai OR (vagy)
 - tagadás
- + össze, arit., centr., adatközlő, tautológia, negáció... stb.

- Egy logikai változót negálva mondunk, ha a változó fölött
vonal van, és pozít, ha nincs.

- Az eredménykombinációs kábelzeteket LOGIKAI KAPUknak hívjuk.



Multiplexer: Olyan kombinációs kábelzet, amelynek 2^n db ^{adat} bemenete,
 n db vezérlőbemenete, 1 db engedélyező bemenet és 1 db
kimenet van.

Demultiplexer: Olyan komb. log kábel, amelynek 1 db adatbemenet, n db
vezérlőbemenet, 2^n db kimenet és 1 db engedélyező bemenet van.

16. tétel

Regiszter: Olyan szerkezet +H-^o-_o, amelyen néhány bit együttes tárolása alkalmas.

SL: shift left: 1 bittel balra \rightarrow 2-vel való szorzás

SR: shift right: 1 bittel jobbra \rightarrow 2-vel való osztás

OE: Output Enable: kimenet engedélyező

IS: Input Serial: soros bemenet

OS: Output Serial: soros kimenet

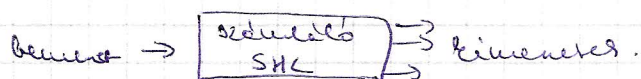
A regiszter korlátja: ideje jóval kisebb, mint a memóriában tárolt információé. A processzor belsőjében lévő regiszter nem memórianevezetű, hanem listabil vibrátor segítségével működik / érvényes.

Szűrő: A SHC egy bizonyos csoportú szűrő. Fellegzettségű, hogy a kimenetűre érkező logikai értékváltozások (impulzusok) szűrték, tárolják.

Impulzus: Két db. értékváltozással határozott jel.

negatív imp.  pozitív imp. 

A szűrő kimeneti értéke csak a belső állapotból függnek. A szűrő egyetlen bemenete az állapotváltás időpontjának meghatározására szolgál.



A szűrőt alkotó elemi tároló működése alapján megkülönböztetünk SZINKRON és ASZINKRON szűrőket.

- A anizónu jelleű névelőe csekben elemi tároló egyűs utáni somendben változtatás állapotát.
- A nízónu jelleű tárolóe csekben az elemi tárolóe állapotváltásáa egyidőben történie. A állapotváltásóe az előző állapotváltás által programozott KFH határozó meg.
- A névelőelés irányá alapján:
 - előre névelőe (növevő somendben nével.)
 - nízónu névelőe (növevő somendben nével.)
 - Ékírású v. reverzibilis névelőe.

anizónu névelőe:

Egyszerűs elérésűtőe elemi somendi tárolóeól kialakított lináris ontóeól.

nízónu névelőe:

nízónu névelőe csekben a tároló áramlóe állapotváltásái egy időben történnek, kékben az anizónu sz-nál gyorsabban működnek. A tárolóe beemekt a nízónu névelőehez tartozó szimulációs logikai helyőat lémeneti érűeie elő.

13. tétel A számítógép funkcionális áramkörei

Az olyan áramköröket, amelyek ugyanazon bináris helyi értékek összegét képezik és kialakítják a szükséges esetben az átvitelt a magasabb helyi értékek felé, de nem veszik figyelembe az előző helyi értéken képződött átvitelt, **félösszeadó áramkör**nek nevezzük.

Az olyan áramköröket, amelyek ugyanazon bináris helyi értékek összegét képezik, kialakítják a szükséges esetben az átvitelt a magasabb helyi értékek felé és figyelembe veszik az előző helyi értéken képződött átvitelt, **teljes összeadó áramkör**nek nevezzük.

A **fél** és **teljes kivonó áramkörök** két szám különbségét képezik. A főbb tulajdonságaik megegyeznek az összeadó áramkörökével.

Tároló áramkörnek azokat az eszközöket nevezzük, amelyek képesek a beléjük táplált információt tetszés szerinti ideig megőrizni. Az ilyen jellegű feladatot a visszacsatolt áramkörök látják el. Az áramkör kimenetén megjelenő jelet fázishelyesen visszavezetik az eszköz bemenetére (pozitív) visszacsatolás. Ezzel a megoldással elérhető, hogy a kimenő jel értéke

- a bemenő jel értékétől,
- a kimenő jel korábbi állapotától,
- ezek sorrendjétől fog függeni.

Az ilyen eszközöket a **szekvenciális** vagy **sorrendi áramkörök** csoportjába sorolják. A **D-típusú tárolókból** épülnek fel a statikus RAM-ok.

Az adatkiválasztó áramkörök közé sorolják az adatszelektort, a multiplexert és a demultiplexert. Az **adatszelektor** feladata, hogy különböző adatok sorozatából a kívánt adatokat kiválassza és a kimeneten keresztül továbbítsa. Azt az áramkört, amely független bemenő jeleket időben egymás után a kimenetre továbbítja, **multiplexer**nek nevezzük. A **multiplexer** időtől függően vezérelt **adatszelektor**. Azt az áramkört, amely a bemenetén megjelenő adatokat adott utasítással az egyik meghatározott kimenetre kapcsolja, **demultiplexer**nek nevezzük.

Igen sok logikai feladatban merül fel igényként, hogy valamilyen kódból egy másik fajta kódot állítsunk elő. Ezt a feladatot a **kódátalakítók** látják el.

A **címdekódoló** több kimenetű áramkör, amelyet különböző elemek vezérelt eléréséhez használnak. Ezt cím segítségével teszi meg. A digitális technikában a **címen** meghatározott hosszúságú bináris jelsorozat, tehát rögzített számú bitből álló bináris szót értünk. Vannak például 2 bites, 4 bites, stb. címek. A címbemenet alapján kiválasztandó kimeneten logikai 1 szint jelenik meg.

Azt az áramkört, amely elvégzi bizonyos események (impulzusok) bekövetkezésének megszámlálását, **számláló áramkör**nek nevezzük (pl.: számítógép programlépéseinek, óraimpulzusainak

15. tétel

Érdelő: A $K \times H$ olyan KH, amelynek u db adatkimentéje, v db adatkimentéje van ($u \geq v$). Az adatkimenték helyzett adatozat a K v db adatkimentéje egy meghatározott szabály szerint más formában adja vissza.

Bináris Gray Érdelő: Egy K v db adatkimentéje és u db adatkimentéje van.

A Gray K v db adatkimentéje rendelkezik, u az egyenest v db adatkimentéje van egy kitéréssel különböznek egymástól. olyan v db adatkimentéje és u db adatkimentéje van, ahol v db adatkimentéje u db adatkimentéje előállításai.

számlálása, címképzés, időzítések végrehajtása, stb.). Az elektronikus számláló áramköröknek igen sok fajtája van. Alapvetően azonban két fő csoportot szoktunk elkülöníteni, az **aszinkron** és a **szinkron** számlálókat. Az aszinkront nagyon sok helyen használják, például minden olyan esetben, amikor frekvencia leosztást kell végezni, azaz nagy rezgésszámból kisebbet kell előállítani. A számítógép belsejében azonban fontos, hogy melyik időpillanatban váltanak állapotot az áramkörök (időzítés), ezért ott szinkron számlálókat használnak.

17. tétel

┌ Azt az áramkört, amely két n bites szó között tetszés szerint összeadást, kivonást, ÉS, VAGY, illetve KIZÁRÓ VAGY műveletet képes elvégezni, **aritmetikai-logikai egységnek (ALU-nak)** nevezzük. Az aritmetikai-logikai egység a számítógép egyik legfontosabb egysége, amely az utasításban előírt műveletet végrehajtja.

Az ALU

- ◆ aritmetikai;
- ◆ logikai;
- ◆ léptető és forgató;
- ◆ bit- és karakterlánc műveletek végrehajtására képes.

A mikroprocesszorok ALU-ja, mint a művelet eszköze, a legegyszerűbb esetben is néhány fő egységet biztosan magában foglal. Ezek a következők:

- ◆ összeadó egység, amely két operandusz összeadására szolgál;
- ◆ léptető áramkörök, amelyek a regiszterek tartalmát művelet végzés közben jobbra, vagybalra léptetik, azaz tulajdonképpen osztják, vagy szorozzák azt;
- ◆ logikai áramkörök a logikai műveletek megoldásához;
- ◆ regiszterek, az adatok ideiglenes tárolására; ezek lehetnek az ALU részét képező, kizárólagos használatú regiszterek, mint pl. többnyire az akkumulátor regiszter (AC), vagy a processzor általános célú regiszterei közül egy, vagy több.

Az aritmetikai műveletek végrehajtásakor, az eredménytől függően az állapotregiszter egyes jelzőbitjeit (flag-jeit) a processzor beállítja. A művelet eredményét visszatükröző legfontosabb jelzőbitek a következők:

- ◆ átvitel (carry) jelzőbitje;
- ◆ nulla (zero) jelzőbitje;
- ◆ előjel (sign) jelzőbitje;
- ◆ túlcsondulás (overflow) jelzőbitje.