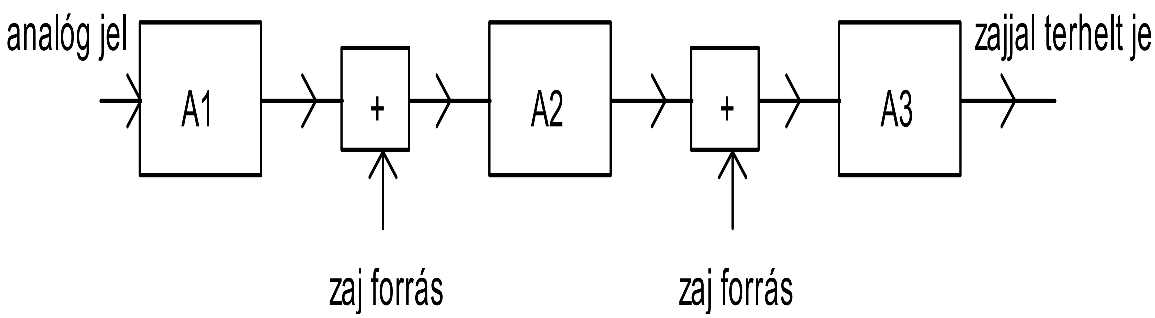
Boole algebra, logikai függvények

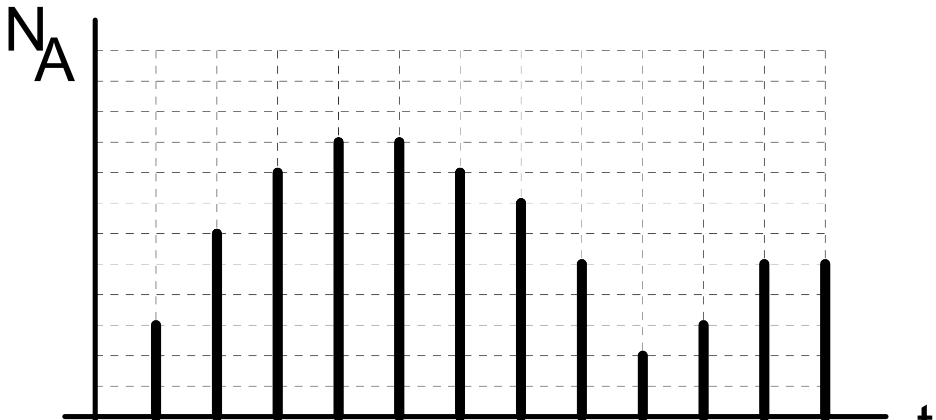
Digitális ↔ analóg

Az átvitel során az analóg jelhez zaj adódik, amelyet szintén felerősít az erősítő. Mennél több analóg jelfeldolgozó áramkörön halad át az analóg jel, annál jobban romlik a jelnek a zajhoz képesti értéke (jel/zaj viszony.) Az analóg jelfeldolgozók méretét ez jelentősen korlátozza.



Információ feldolgozó digitális gép

analóg jel



digitalizált jel (mintavételezés, kvantálás)

Az információ feldolgozó gép kódokkal dolgozik

analóg kód: amplitúdóban és időben folytonos

digitális kód*:* amplitúdóban és időben diszkréten értelmezett (véges idő alatt véges információt hordoz)

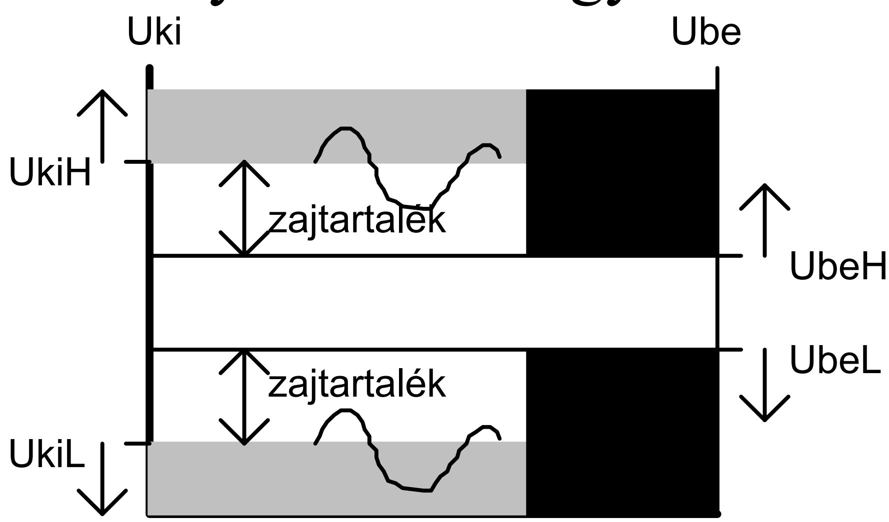
Analóg jelből digitális jel:

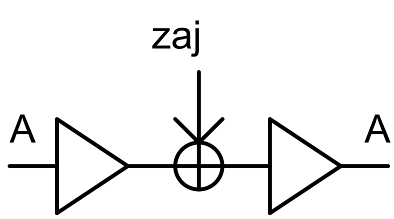
- ***mintavételezéssel*** (adott időpontban mekkora a jel analóg értéke) majd
- ***kvantálással*** (a mintavett analóg érték *n* bites digitális számmá alakításával) nyerhető.

A digitalizált információ az analógnál *sok nagyságrenddel pontosabban* vihető át, ill. dolgozható fel.

A digitális információ helyes átvitelét egyrészt áramkörileg az analóghoz képest sokkal nagyobb biztonsággal meg lehet oldani, mert 1 bitnyi információ továbbításakor csak 2 érték fordulhat elő. Pl. a 0-át és az 1-et reprezentáló feszültség vagy áram érték viszonylag nagy zaj mellet is jól megkülönböztethető megfelelő áramköri megoldásokkal.

A logikai értéket feszültség tartományok reprezentálják.
A tartományok átfedik egymást:





A bemeneti jelre szuperponálódó kismértékű zaj még nem okoz hibás logikai szint érzékelést!

Megfelelő kódolással az előforduló hibákat jelezni, sőt javítani is lehet.

Boole algebra

Boole, George (1815. - 1864.) Angol matematikus. A logikai algebra), amelyet **Boole algebrának** nevezünk, megteremtésével a számítástechnika fejlődéséhez jelentősen hozzájárult.

Boole algebra ($\mathcal{B}$)

$\mathcal{B} = < B, +, *, \overline{} >$ (Boole *halmaz*, logikai VAGY, logikai ÉS, invertálás *műveletek*)

B={0, 1, α, β, γ,..)} logikai konstansok és változók.

Csak két érték létezik:
0: hamis
1: igaz

(Létezik 2-nél több értékű logika is.)

A Boole algebra a Boole halmazon végezhető műveleteket adja meg **axiómákkal** (műveleti szabályok) és **definíciókkal.**

Axiómák:

Kommutativitás (felcserélhetőség):

A1. $(\alpha+\beta) = (\beta + \alpha)$

A2. $\alpha.\beta = \beta.\alpha$

Disztributivitás:

A3. $\alpha.(\beta+\gamma) = (\alpha.\beta) + (\alpha.\gamma)$

A4. $\alpha + (\beta.\gamma) = (\alpha + \beta)(\alpha + \gamma)$

Műveletek a konstansokkal:

A5. $\alpha + 0 = \alpha$

A6. $\alpha.1 = \alpha$

D: **Inverz elem** ($\overline{\alpha}$) **definíciója**:
(Ezt a szövegszerkesztő miatt a továbbiakban /α-al jelöljük.)

Létezik minden α-hoz egyérteműen /α, melyre:

D1: $\alpha./\alpha = 0$, D2: $\alpha + /\alpha = 1$

A fentiekből minden Boole algebrai azonosság ellentmondás mentesen levezethető.

Dualitás elve:

A + ↔ *, és 0 ↔ 1 felcserélésével is igazak maradnak a Boole algebrai azonosságok.

Pl: A3->A4 (disztributivitás)

A5->A6 műveletek konstansokkal

A Boole algebra fontos tulajdonságai (nem bizonyítjuk):

T1. $\alpha + \alpha = \alpha$ T2. $\alpha.\alpha = \alpha$

T3. $\alpha + 1 = 1$ T4. $\alpha.0 = 0$

elnyelés:

T5. $\alpha + \alpha.\beta = \alpha$ T6. $\alpha.(\alpha + \beta) = \alpha$

T7. $/1 = 0$ T8. $/0 = 1$

T9. $// \alpha = \alpha$

Asszociativitás (csoportosíthatóság):

T10. $\alpha+(\beta + \gamma) = (\alpha + \beta)+\gamma$

T11. $\alpha.(\beta.\gamma) = (\alpha.\beta).\gamma$

De' Morgan azonosságok:

T13. $/(\alpha + \beta) = /\alpha./\beta$ $/\Sigma\, \alpha_i = \Pi\, /\alpha_i$

T14. $/(\alpha.\beta) = /\alpha + /\beta$ $/\Pi\, \alpha_i = \Sigma\, /\alpha_i$

Példa:

Hozzuk egyszerűbb alakra!

1. $Y1 = /A + A.B$

Megoldás: A4 és D2 alalpján

$/A + A.B = (/A + A)(/A + B) =$

$= 1. (/A + B) = /A + B$

2. $Y2 = /A + A.B./C +$

$+ (/A + A.B./C)(/A + /A./B.C)$

Megoldás:

$/A + A.B./C = X$ helyettesítéssel, az elnyelés miatt, majd 1. alapján:

$Y2 = X + X. (/A + /A./B.C) = X =$

$= /A + A.(B./C) = /A + B./C$

Szöveges logikai feladat:

Tengerparton pihen Apa, Mama,Fiuk, L1 lányuk, L2 lányuk.

1. Ha az Apa úszik, akkor a Mama és Fiúk is megy.
2. Ha a Fiú úszik, akkor L1 nővére is.
3. L2 lány pontosan akkor úszik, ha a Mama is.
4. Minden reggel úszik valamelyik szülő.
5. Vasárnap reggel csak az egyik lány úszott.

Kik úsztak vasárnap reggel?

Logikai egyenletek a szöveg alapján:

1. Ha az Apa úszik, akkor a Mama és Fiúk is megy.

A.M.F + /A = 1 (igaz állítás)

2. Ha a Fiú úszik, akkor L1 nővére is.

F.L1 + /F = 1

3. L2 lány pontosan akkor úszik, ha a Mama is

L2.M + /L2./M = 1

4. Minden reggel úszik valamelyik szülő.

A + M = 1

5. Vasárnap reggel csak az egyik lány úszott.

L1./L2 + /L1.L2 = 1

Igaz állítások logikai szorzata is igaz.

Szorozzuk össze az azonosságokat, olyan sorrendben, hogy közben mennél több tag kiessen!

1, 4: (AMF + /A)(A + M) = AMF + AMF +/A.A + /A.M = AMF + /AM

3: (AMF + /AM)(ML2 + /M/L2) =

= AMFL2 +/AML2

5:(AMFL2+/AML2)(L1/L2 +/L1L2)=

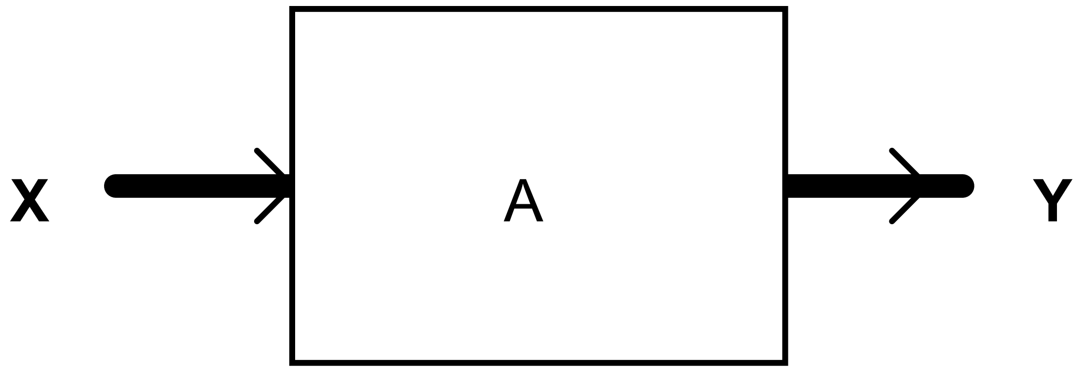
= AMFL2/L1 + /AML2/L1

2:(AMFL2/L1+/AML2/L1)(FL1+/F)=

=/AML2/L1/F

Tehát a Mama és L2 lánya úszott.

A digitális automata



X: $(x_1, x_2 .. x_n)$ bemeneti logikai változók

Y: $(y_1, y_2 \ldots y_m)$ kimeneti logikai változók

X*, Y*: bemeneti (kimeneti) sorozat, a bemenet (kimeneti) változók időben egymást követő értékei.

A digitális automata az

X* → Y* leképzést állítja elő.

Sorrendi automata:

$Y^t = g(X^t,Q^t)$ Q^t : állapot, az emlékező logikai elemek (flip-flopok) kimeneteinek t időpontbeli értéke.

Az aktuális kimenet az aktuális bemenet és aktuális állapot függvénye.

$Q^{t+1}=f(X^t, Q^t)$

A következő állapot az aktuális bemenet és az aktuális állapot függvénye (rekurzív, önhivatkozó megadás).

A kimenet a bemenetet ért változások sorrendjétől is függ.

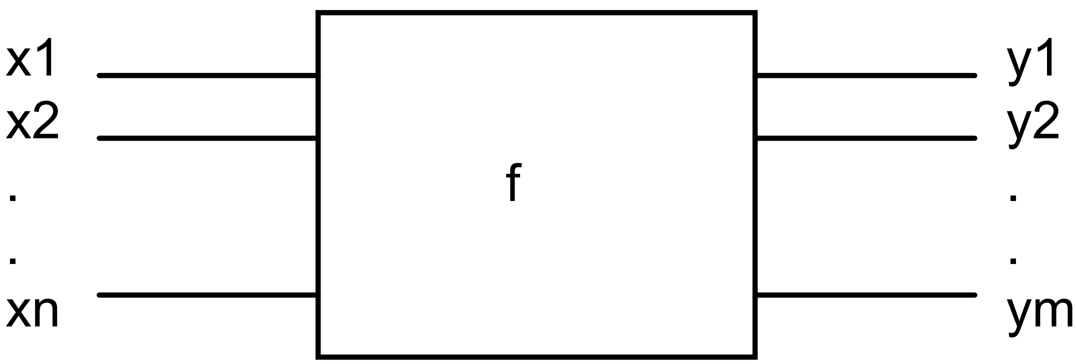
Kombinációs automata

$Y^t = g(X^t)$

***Az aktuális kimenet csak az aktuális bemenettől függ*.**

Logikai függvények

n bementű *m* kimenetű log. fgv.:



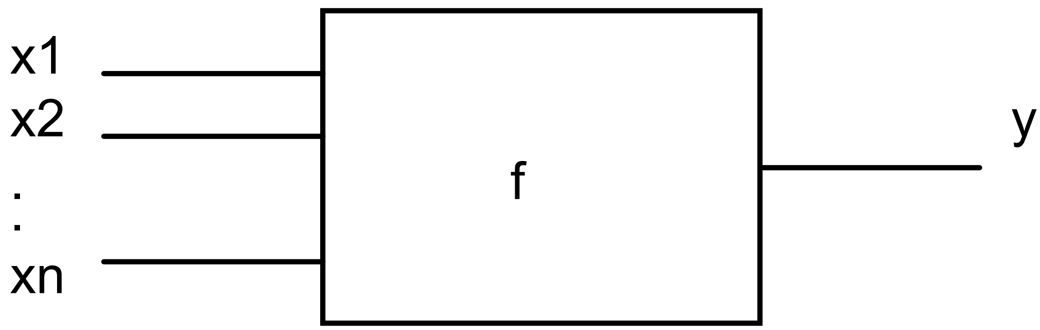
$y_1=f_1(x_1..x_n)$

$y_2= f_2(x_1..x_n)$

...

$y_m= f_m(x_1..x_n)$

Az előbbi tulajdonképpen m db. n bemenetű logikai függvény.



Logikai függvények megadása:

- táblázatos formában, ***igazságtáblával***

- ***általános Boole alg. alak*** (tetszőleges Boole kifejezés) pl.

f= A + B(C+/D)

- ***normál (kanonikus) alakok*** (*diszjunktív normál alak, konjunktív normál alak*) lsd. később

- **kapcsolási rajzzal** (lsd. később)

Igazságtábla (a bementekhez megadjuk a kimeneteket)

Az összes 2 változós (A, B) függvény igazságtáblája:

AB	f_0^2	f_1^2	f_2^2	f_3^2	f_4^2	f_5^2	f_6^2	f_7^2	f_8^2	f_9^2	f_{10}^2	f_{11}^2	f_{12}^2	f_{13}^2	f_{14}^2	f_{15}^2
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Az *n* változós logikai függvények száma:

Sorok száma: $s=2^n$ Oszlop kitöltéseinek száma: $N=2^s$

Tehát $N = 2^{(2^\wedge n)}$

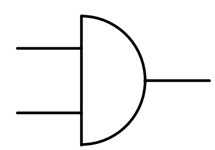
Az összes n változós m kimenetű logikai függvények száma: $N = (2^{(2\wedge n)})^m = 2^{m(2\wedge n)}$

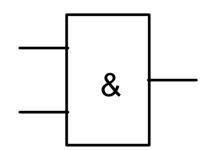
A kétváltozós függvények, nevezetes kapuk.

AB	$\mathbf{f_0^2}$	$\mathbf{f_1^2}$	f_2^2	f_3^2	f_4^2	f_5^2	f_6^2	f_7^2	f_8^2	f_9^2	f_{10}^2	f_{11}^2	f_{12}^2	f_{13}^2	f_{14}^2	$\mathbf{f_{15}^2}$
00	**0**	**0**	0	0	0	0	0	0	1	1	1	1	1	1	1	**1**
01	**0**	**0**	0	0	1	1	1	1	0	0	0	0	1	1	1	**1**
10	**0**	**0**	1	1	0	0	1	1	0	0	1	1	0	0	1	**1**
11	**0**	**1**	0	1	0	1	0	1	0	1	0	1	0	1	0	**1**

$f_0^2 = 0$ $\quad$ $f_{15}^2 = 1$ logikai konstansok

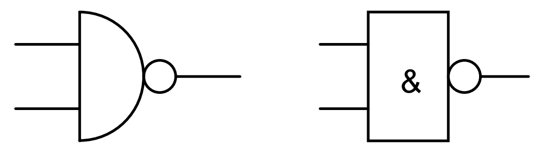
$f_1^2 = AB$ $\quad$ **AND**, **(ÉS)** kapu rajzjele:





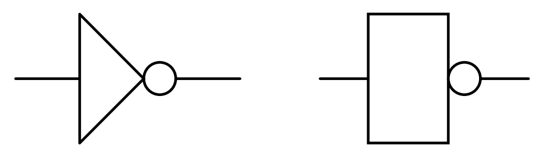
AB	f_0^2	f_1^2	$\mathbf{f_2^2}$	$\mathbf{f_3^2}$	$\mathbf{f_4^2}$	f_5^2	f_6^2	f_7^2	f_8^2	f_9^2	f_{10}^2	$\mathbf{f_{11}^2}$	$\mathbf{f_{12}^2}$	$\mathbf{f_{13}^2}$	$\mathbf{f_{14}^2}$	f_{15}^2
00	0	0	**0**	**0**	**0**	0	0	0	1	1	1	**1**	**1**	**1**	**1**	1
01	0	0	**0**	**0**	**1**	1	1	1	0	0	0	**0**	**1**	**1**	**1**	1
10	0	0	**1**	**1**	**0**	0	1	1	0	0	1	**1**	**0**	**0**	**1**	1
11	0	1	**0**	**1**	**0**	1	0	1	0	1	0	**1**	**0**	**1**	**0**	1

f_{14}^2 = /(AB) = /A + /B **NAND, (NEM ÉS)**



f_2^2 = A./B f_{13}^2 = /(A./B) = /A + B

f_3^2= A f_{12}^2= /A **INVERTER**

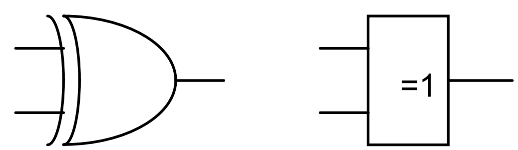


f_4^2 = /A.B f_{11}^2 =/(/AB)=A+/B)

AB	f_0^2	f_1^2	f_2^2	f_3^2	f_4^2	$\mathbf{f_5^2}$	$\mathbf{f_6^2}$	f_7^2	f_8^2	f_9^2	$\mathbf{f_{10}^2}$	f_{11}^2	f_{12}^2	f_{13}^2	f_{14}^2	f_{15}^2
00	0	0	0	0	0	**0**	**0**	0	1	1	**1**	1	1	1	1	1
01	0	0	0	0	1	**1**	**1**	1	0	0	**0**	0	1	1	1	1
10	0	0	1	1	0	**0**	**1**	1	0	0	**1**	1	0	0	1	1
11	0	1	0	1	0	**1**	**0**	1	0	1	**0**	1	0	1	0	1

$f_5^2 = B \quad f_{10}^2 = /B$ **(INVERTER)**

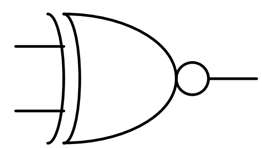
$f_6^2 = A./B + /A.B = A\ (+)\ B$ **EXOR**, **(KIZÁRÓ VAGY, MODULO2 összeadás)**

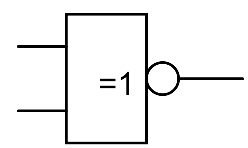


AB	f_0^2 0	f_1^2 AND	f_2^2	f_3^2	f_4^2	f_5^2	f_6^2 EXOR	f_7^2 OR	f_8^2 NOR	$\mathbf{f_9^2}$ **EKV**	f_{10}^2 INV.	f_{11}^2	f_{12}^2 INV.	f_{13}^2	f_{14}^2 NAND	f_{15}^2 1
00	0	0	0	0	0	0	0	0	1	**1**	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	**0**	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	**0**	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	**1**	0	1	0	1	0	1

$$f_9^2 = A.B + /A./B = A\ (\bullet)\ B =$$

$$= (A+/B)(/A+B)$$

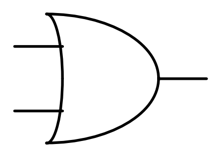


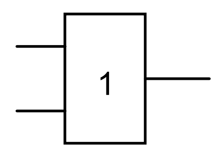


EKVIVALENCIA

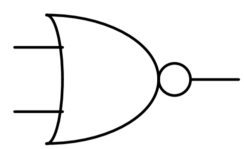
AB	f_0^2 0	f_1^2 AND	f_2^2	f_3^2	f_4^2	f_5^2	f_6^2 EXOR	**f_7^2 OR**	**f_8^2 NOR**	f_9^2 EKV	f_{10}^2 INV.	f_{11}^2	f_{12}^2 INV.	f_{13}^2	f_{14}^2 NAND	f_{15}^2 1
00	0	0	0	0	0	0	0	**0**	**1**	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	**1**	**0**	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	**1**	**0**	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	**1**	**0**	1	0	1	0	1	0	1

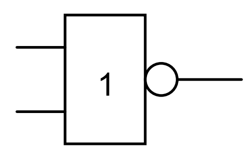
$f_7^2 = A + B$ **OR, (VAGY)**





$f_8^2 = /(A + B) = /A./B$ **NOR**

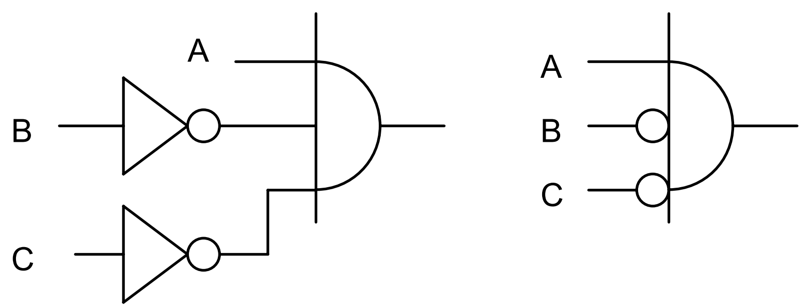




Elterjedt jelöléstechnika a CAD rendszereknél:

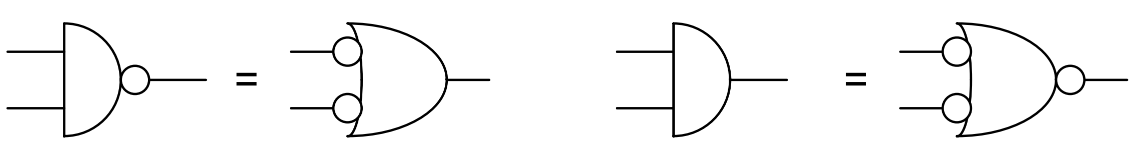
A karika invertálást jelent, egy kapu tetszőleges bemenetén is használható.

Pl. A./B./C

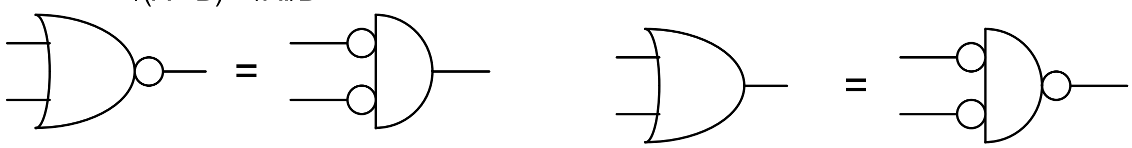


A De' Morgan azonosságok alapján az egyes kapukat másként is rajzolhatjuk:

/(A.B) = /A + /B



/(A+ B) = /A./B



Normál (kanonikus) alakok

Diszjunktív normál alak

Szorzatok összege (sum of product), *a szorzat tagokban minden változó szerepel (nem egyszerűsített alak)*. Az egyes tagok az igazságtábla 1-eseti valósítják meg, ezek a ***mintermek***:

Pl. $f_7^2 = /A.B + A./B + A.B$

AB	f_7^2 OR
00	0
01	1
10	1
11	1

Általánosan:

$f = \Sigma\, \alpha_i . m_i \quad \alpha_i : \{0, 1\}$

$f = 0.m^2{}_0 + 1.m^2{}_1 + 1.m^2{}_2 + 1.m^2{}_3 =$

$= m^2{}_1 + m^2{}_2 + m^2{}_3 = \Sigma^2\ 1, 2, 3$

A term index meghatározása: termek változói ABC sorrenben írjuk fel, a ponált változóhoz 1-et, a negálthoz 0-át rendelünk, majd az így kapott bináris számot decimálisan kilvassuk: pl. $A/B \rightarrow 10_B \rightarrow 2_D$

Konjunktív normál alak

Összegek szorzata (product of sum), *az összeg tagokban minden változó szerepel (nem egyszerűsített alak)*. Az egyes tagok az igazságtábla 0-áit valósítják meg, ezek a ***maxtermek***:

Pl. $f_2^2 = (/A+/B)\ (A+/B)\ (A+B)$

AB	f_2^2 OR
00	0
01	0
10	1
11	0

Általánosan:

$f = \Pi\ (\beta_i + M_i) \quad \beta_i : \{0, 1\}$

$f = (0 + M^2_0)(1+M^2_1)(0+ M^2_2)(0+M^2_3) = M^2_0\ M^2_2\ M^2_3$

$= \Pi^2\ 0, 2, 3$

Az index mint a diszj. alaknál: pl. $(A + /B) \rightarrow 10_B \rightarrow 2_D$

Átalakítás a két normál alak között:

Legyen f a következő 3 változós függvény: $f = \Sigma^3\ 0, 2, 3, 4, 6, 7$

Ennek a negáltjában azok mintermek szerepelnek, melyek f-ből hiányoznak. $/f = \Sigma\ 1, 5$

$f = //f =/(\Sigma\ 1, 5) = /(m1 + m5) =$

$= /(/A./B.C + A./B.C) =$

$= /(/A./B.C)./(A./B.C)=$

$= (A+B+/C)(/A+B+/C) = M^2{}_6 M^2{}_2$

Minden logikai függvénynek pontosan 1 diszjunktív és egy konjunktív normál alakja van.

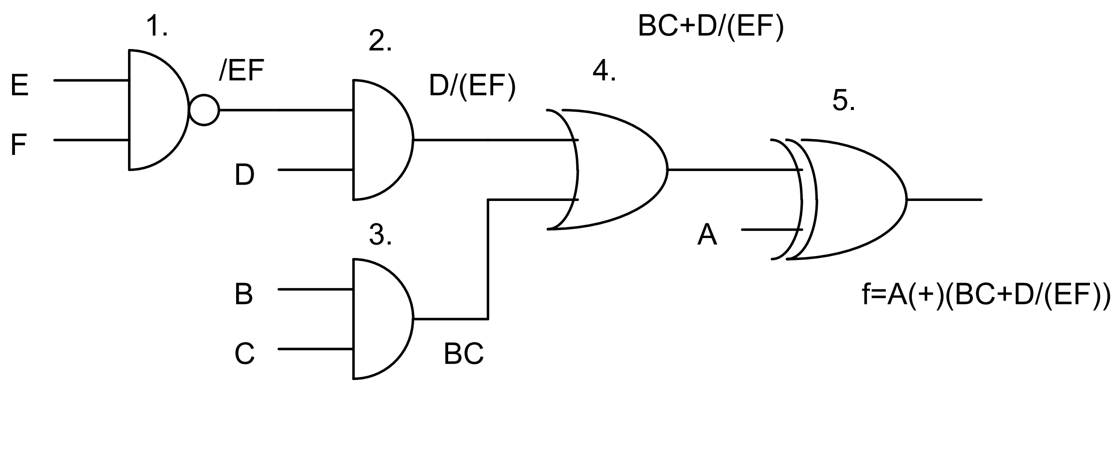
(Az igazságtáblából történő felírásból triviális.)

Logikai függvény megadása kapcsolási rajzzal

Az általános algebrai alakból a felírás struktúráját megőrző *közvetlen felrajzolás*:

Legyen f = A (+) (BC +D/(EF))

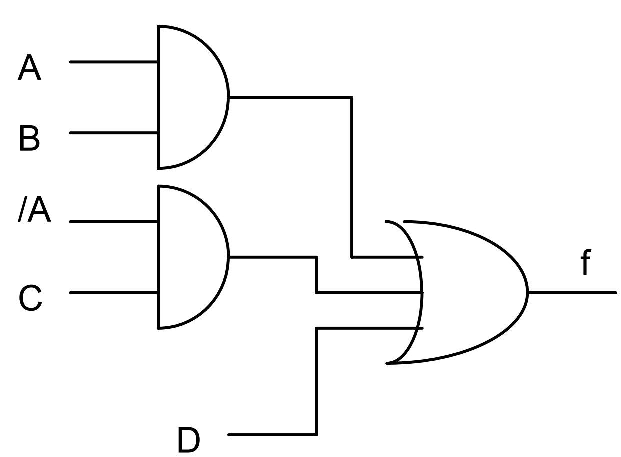
Először a legbelső zárójelet valósítjuk meg, majd fokozatosan az egyre kijjebb levőket. A sorrendet a kapuk felé írtuk.



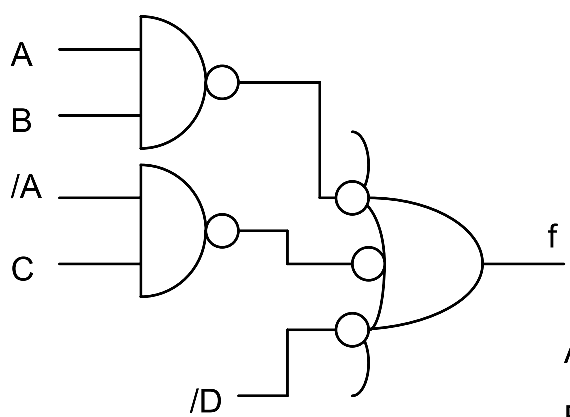
Kapcsolási rajzot sokszor az egyszerűsített (nem normál) diszjunktív ill, konjunktív alak alapján rajzolunk. Itt, mivel már nem biztos, hogy az egyes tagokban minden változó szerepel, már nem min – vagy maxtermekről, hanem csak termekről beszélünk.

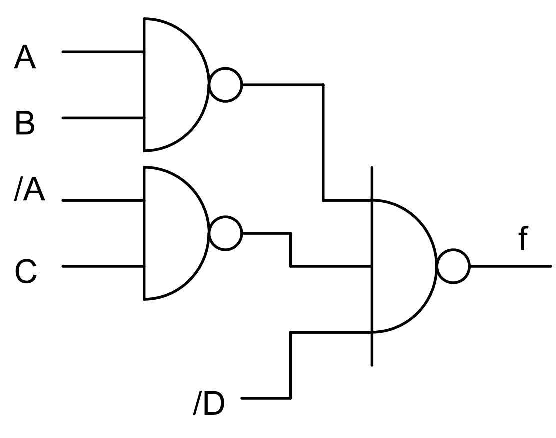
Pl. Az f = AB + /AC + D egy diszjunktív (mivel szozatok összege), de nem normál megadás.

Az ehhez tartozó közvetlen megvalósítás:



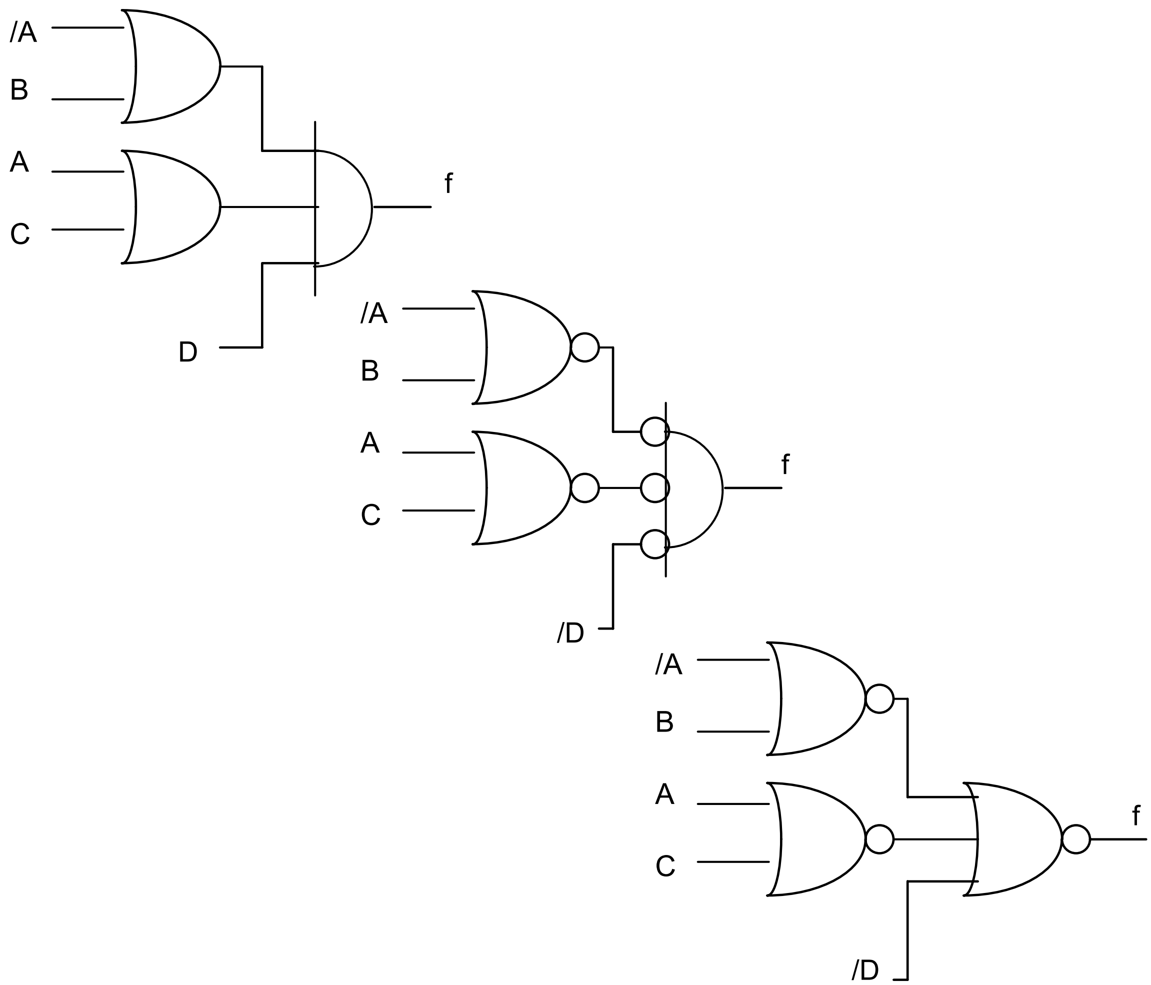
A diszjunktív megadású függvényt csak NAND kapukkal is megvalósíthatjuk. Ezt nevezzük *homogén NAND* kapus megvalósításnak. Az összes ÉS kapu kimenetére és a VAGY minden bemenetére invertert téve, a 2 sorbakötött inverter a logikán nem változtat, ugyanakkor minden kapu NAND-dá válik. (Figyeljük meg, hogy a D-t is meg kell invertálni!)



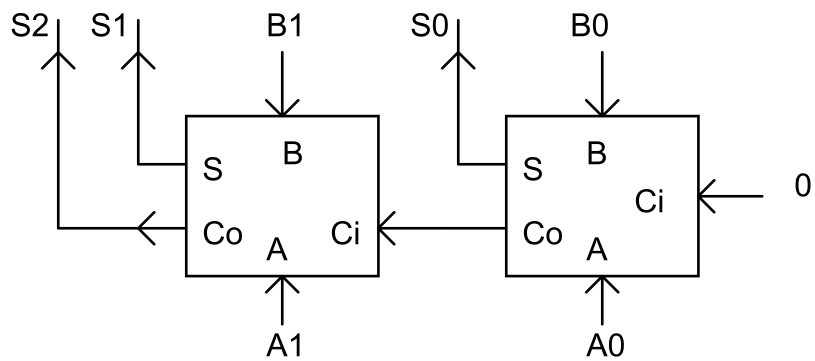


Az előzőkhöz hasonlóan a konjunktív megadású logikai függvények csak NOR kapukkal is megvalósíthatók. (Homogén NOR realizálás.)

Legyen f = (/A+B)(A+C)D



Tervezzünk 1 bites bővíthető összea-dót:



A	B	Ci	So	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

So = /A/BCi +

+/AB/Ci +

+A/B/Ci

+ABCi

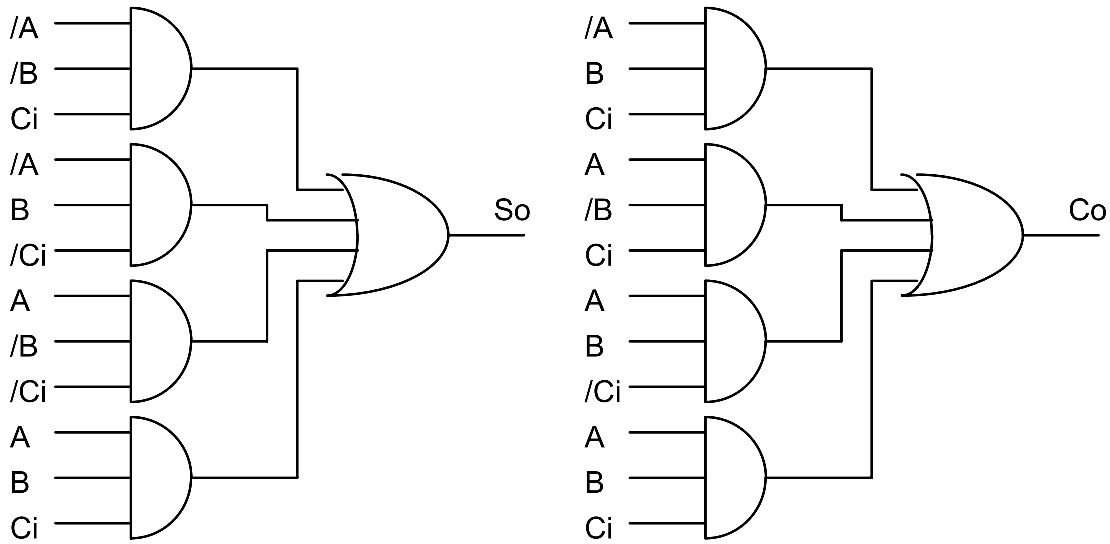
Co = /ABCi+

+ A/BCi+

+ AB/Ci +

+ ABCi

Az összeadó kapcsolási rajza:

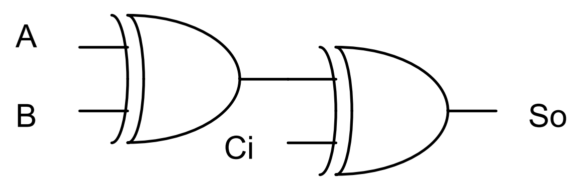


Az S olyan speciális függvény, mely a bemenetein levő 1db és 3 db 1-es esetén ad 1-et. Az olyan fgv.-eket, melyek kimenete csak a bementükön levő 1-esek számától függ, ***szimmetrikus fgv.***-nek nevezik.

So = $S^3_{1,3}$ Alsó index az 1-esek száma, amire a kimenet is 1-et ad, a felső index a változók száma.

Az EXOR kapu: A./B +/A.B = S^2_1

Az EXOR kapu So-hoz hasonlóan páratlan számú 1-es esetén ad 1-et. Ezért So csak EXOR kapukkal megvalósítható:



Az EXOR kapu vezérelhető inverter-ként is használható: