# DIGITÁLIS TECHNIKA
# 3

Előadó:
Dr. Oniga István

# Logikai függvények

- A logikai függvény olyan egyenlőség, amely változói kétértékűek, és ezek között csak logikai műveleteket végzünk
- A függvények megadása történhet
  - táblázat segítségével (Igazságtábla),
  - algebrai alakban,
  - matematikai jelölésekkel,
  - grafikus módon (Karnaugh tábla),
  - időfüggvény formájában.
- A felsorolt leírási módok teljesen egyenértékűek, és egymásba átírhatók!

- Az igazságtáblázatban a független változók lehetséges állapotait bináris kód szerint soroljuk fel. Egy n számú független változójú logikai függvény $2^n$ féle állapotot vehet fel.

# Igazságtáblázat

- A táblázat tartalmazza a független változók összes kombináció-ját és az azokhoz rendelt *függvényértékét*
- A független változók lehetséges állapotait bináris kód szerint soroljuk fel. Egy n számú független változójú logikai függvény $2^n$ féle állapotot vehet fel.
- A felsorolás kétféle lehet:
  - logikai szorzatokra, azaz mintermek,
  - logikai összegekre, azaz maxtermek.
- Minterm = olyan logikai ÉS, melyben minden változó egyszer, és csakis egyszer fordul elő negált, vagy nem negált formában.
- Maxterm = olyan logikai VAGY, melyben minden változó egyszer, és csakis egyszer fordul elő negált, vagy nem negált formában.
- Egy n változós logikai függvénynek tehát $2^n$ féle mintermje és ugyanennyi maxtermje lehet.

| Minterm | Maxterm |
|---|---|
| $\overline{B}\,\overline{A}$ | $B + A$ |
| $\overline{B}A$ | $B + \overline{A}$ |
| $B\overline{A}$ | $\overline{B} + A$ |
| $BA$ | $\overline{B} + \overline{A}$ |

| Minterm | Maxterm |
|---|---|
| $\overline{C}\,\overline{B}\,\overline{A}$ | $C + B + A$ |
| $\overline{C}\,\overline{B}A$ | $C + B + \overline{A}$ |
| $\overline{C}B\overline{A}$ | $C + \overline{B} + A$ |
| $\overline{C}BA$ | $C + \overline{B} + \overline{A}$ |
| $C\overline{B}\,\overline{A}$ | $\overline{C} + B + A$ |
| $C\overline{B}A$ | $\overline{C} + B + \overline{A}$ |
| $CB\overline{A}$ | $\overline{C} + \overline{B} + A$ |
| $CBA$ | $\overline{C} + \overline{B} + \overline{A}$ |

# Logikai függvények felírása

Két normál forma:

- Diszjunktív kanonikus alak (teljes diszjunktív normál forma): olyan függvény, mely mintermek VAGY kapcsolatából áll.
  - Sum of Products (SOP)
  - Szokás még egyszerűen mintermes alaknak, szorzatok összegének, röviden szorzatösszegnek, mintermek összegének is nevezni.
  - Pl. $$X = \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C} + ABC$$

- Konjunktív kanonikus alak, (teljes konjunktív normál forma): olyan függvény, mely maxtermek ÉS kapcsolatából áll.
  - Product of Sums (POS)
  - Szokás még egyszerűen maxtermes alaknak, összegek szorzatának, röviden összegszorzatnak, maxtermek szorzatának is nevezni.
  - Pl. $$X = (\overline{A} + \overline{B} + C)(\overline{A} + \overline{B} + \overline{C})(A + B + C)$$

# Matematikai, egyszerűsített felírás

- A függvény, oly módon is leírható, hogy a minterm-je vagy maxterm-je hanyadik eleme a mintermek, illetve maxtermek rendezett sorának.
- A minterm-et az $m_i^v$ jelöléssel helyettesíthetjük,
  - m jelzi, hogy a logikai egység minterm,
  - a felső index v a változók számát,
  - az alsó index i a sorszámot jelenti.
- Súlyozzuk a következőképen egy háromváltozós függvény változóit:

$$C - 2^2,\ B - 2^1,\ A - 2^0,$$

Az előző példák a következő képen írhatók át:

$$X = \overline{C}\,\overline{B}A + \overline{C}\,\overline{B}\,\overline{A} + CBA = m_1^3 + m_0^3 + m_7^3 = P_1 + P_0 + P_7$$

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

$$= M_1^3 \bullet M_0^3 \bullet M_7^3 = S_1 \bullet S_0 \bullet S_7$$

| Minterm | Maxterm |
|---|---|
| $m_0^2 = \overline{B}\,\overline{A}$ | $M_3^2 = B + A$ |
| $m_1^2 = \overline{B}A$ | $M_2^2 = B + \overline{A}$ |
| $m_2^2 = B\overline{A}$ | $M_1^2 = \overline{B} + A$ |
| $m_3^2 = BA$ | $M_0^2 = \overline{B} + \overline{A}$ |

| Minterm | Maxterm |
|---|---|
| $m_0^3 = \overline{C}\,\overline{B}\,\overline{A}$ | $M_7^3 = C + B + A$ |
| $m_1^3 = \overline{C}\,\overline{B}A$ | $M_6^3 = C + B + \overline{A}$ |
| $m_2^3 = \overline{C}B\overline{A}$ | $M_5^3 = C + \overline{B} + A$ |
| $m_3^3 = \overline{C}BA$ | $M_4^3 = C + \overline{B} + \overline{A}$ |
| $m_4^3 = C\overline{B}\,\overline{A}$ | $M_3^3 = \overline{C} + B + A$ |
| $m_5^3 = C\overline{B}A$ | $M_2^3 = \overline{C} + B + \overline{A}$ |
| $m_6^3 = CB\overline{A}$ | $M_1^3 = \overline{C} + \overline{B} + A$ |
| $m_7^3 = CBA$ | $M_0^3 = \overline{C} + \overline{B} + \overline{A}$ |

# Grafikus ábrázolás

- A grafikus ábrázolásainak egyik változata – Karnaugh diagram
- Az függvényt cellákból álló táblán ábrázoljuk. Minden cella egy-egy termet képvisel
- A logikai függvényt úgy írjuk a Karnaugh táblába, hogy amelyik termje 1, az annak a termnek megfelelő cellába 1-et írunk, a többi cellát üresen hagyjuk, azaz a 0-kat nem írjuk be

*Két változós Karnaugh diagram*

| B \ A | 0 | 1 |
|---|---|---|
| 0 | $m_0$ | $m_1$ |
| 1 | $m_2$ | $m_3$ |

*Háromváltozós Karnaugh tábla*

A három változó már $2^3 = 8$ lehetséges állapotot vehet fel, így 8 cellás Karnaugh tábla kell

| C \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| 1 | $m_4$ | $m_5$ | $m_7$ | $m_6$ |

| | | A | A | |
|---|---|---|---|---|
| | 000 | 001 | 011 | 010 |
| C | 100 | 101 | 111 | 110 |
| | | | B | B |

# Négy változós Karnaugh tábla

A négy változó már $2^4 = 16$ lehetséges állapotot vehet fel, így 16 cellás Karnaugh tábla kell

| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| 01 | $m_4$ | $m_5$ | $m_7$ | $m_6$ |
| 11 | $m_{12}$ | $m_{13}$ | $m_{15}$ | $m_{14}$ |
| 10 | $m_8$ | $m_9$ | $m_{11}$ | $m_{10}$ |

B

| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0000 | 0001 | 0011 | 0010 |
| 01 | 0100 | 0101 | 0111 | 0110 |
| 11 | 1100 | 1101 | 1111 | 1110 |
| 10 | 1000 | 1001 | 1011 | 1010 |

C

D

A

# *A logikai függvények egyszerűsítése*

- Cél: *kevesebb **művelet,** és vagy kevesebb változó.*

- *Az* egyszerűsítésre azért van szükség, mert a feladatot megvalósító logikai hálózat kevesebb áramkört, vagy kevesebb utasítást tartalmaz
- *Algebrai módszerek*
- *Grafikus **egyszerűsítés***.
  - A grafikus függvényegyszerűsítés szabályi:
    - az összevont  cellákat lefedő hurokkal szokás jelölni
    - két szomszédos cellában 1 van, akkor ezek összevonhatók (az a változó amelyikben különböznek a cellák kiesik)
    - a összevonható cellák száma $2^n$, ha azok kölcsönösen szomszédosak
    - mindig a lehető legnagyobb tömböt célszerű kialakítani
    - tábla szélei is szomszédosak egymással
    - átlósan nem lehet összevonni
    - minden 1 -t tartalmazó cellát legalább egyszer le kell fedni
    - a lefedett cellákból a kitevőnek (n) megfelelő számú változó esik ki, amelyek a lefedés alatt változnak,

# Minimalizálási algoritmusok

- Sok bemeneti változóra a kézi módszerek (algebrai, Karnaugh tábla) már nem megfelelőek
- Léteznek számítógépes algoritmusok
  - Quine-McCluskey
    - Kimerítő teljes algoritmus
    - Közepes változó számig
    - A megismert lépéseket hajtja végre
    - Számítógépes végrehajtás (hosszú futásidőt igényel)
  - Espresso
    - Heurisztikus algoritmus, lokális keresést alkalmaz
    - Szinte minden szintézis program ezt használja (A Xilinx ISE XST is)

# *Részben meghatározott logikai függvények egyszerűsítése*

- Ha a logikai függvény nem teljesen határozott, akkor legalább egy olyan bemeneti kombináció, (minterm) van, amelyhez rendelt függvény érték számunkra közömbös.
- A közömbös mintermekkel szabadon bánhatunk.
- Ha előnyös az egyszerűsítés szempontjából, akkor összevonjuk őket az ***1-***es mintermekkel, ha nem, akkor ***0***-s mintermeknek tekintjük őket.

Pl.

$$F = \overline{D}\bullet\overline{C}\bullet\overline{B}\bullet\overline{A} + \overline{D}\bullet\overline{C}\bullet B\bullet\overline{A} + \overline{D}\bullet\overline{C}\bullet B\bullet A +$$

$$+ \overline{D}\bullet C\bullet\overline{B}\bullet A + D\bullet\overline{C}\bullet B\bullet\overline{A} + D\bullet C\bullet B\bullet A$$

- Közömbös mintermekk:

$$\overline{D}\bullet C\bullet B\bullet A \qquad D\bullet\overline{C}\bullet\overline{B}\bullet\overline{A} \qquad D\bullet\overline{C}\bullet B\bullet A$$

F = $P_0 + P_2 + P_3 + P_5 + P_{10} + P_{15}$
közömbös mintermekk: $P_7$, $P_8$ şi $P_{11}$

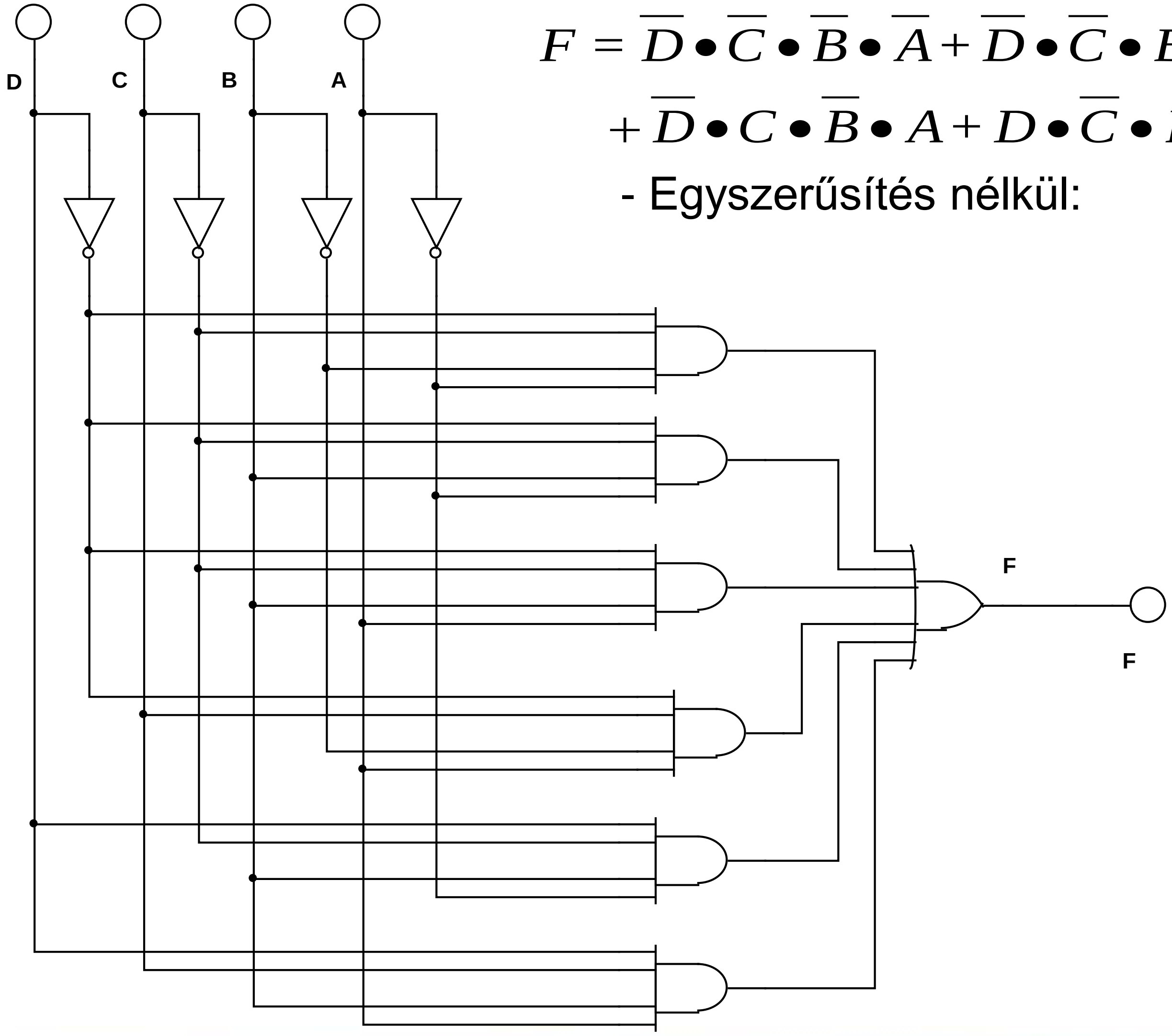
| D | C | B | A | F |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | X |
| 1 | 0 | 0 | 0 | X |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

# A függvény megvalósítassa kapukkal

$$F = \overline{D} \bullet \overline{C} \bullet \overline{B} \bullet \overline{A} + \overline{D} \bullet \overline{C} \bullet B \bullet \overline{A} + \overline{D} \bullet \overline{C} \bullet B \bullet A +$$

$$+ \overline{D} \bullet C \bullet \overline{B} \bullet A + D \bullet \overline{C} \bullet B \bullet \overline{A} + D \bullet C \bullet B \bullet A$$

- Egyszerűsítés nélkül:



# A függvény *egyszerűsítése*

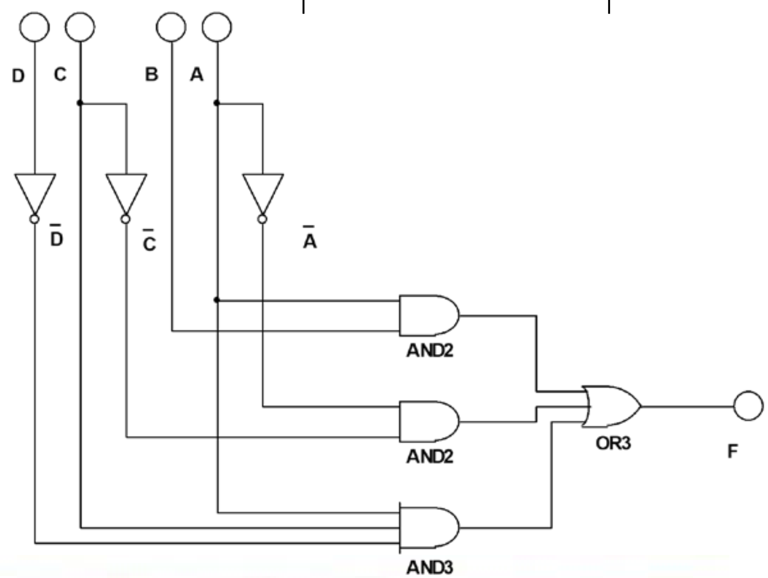
$$F = \overline{D}\bullet\overline{C}\bullet\overline{B}\bullet\overline{A} + \overline{D}\bullet\overline{C}\bullet B\bullet\overline{A} + \overline{D}\bullet\overline{C}\bullet B\bullet A +$$
$$+ \overline{D}\bullet C\bullet\overline{B}\bullet A + D\bullet\overline{C}\bullet B\bullet\overline{A} + D\bullet C\bullet B\bullet A$$

| DC \ BA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 1 | 1 |
| 01 | 0 | 1 | X | 0 |
| 11 | 0 | 0 | 1 | 0 |
| 10 | X | 0 | X | 1 |

- Karnaugh táblázat
- Cellák összevonása
- Egyszerűsítet függvény kiolvasása

$$F = A\bullet B + \overline{A}\bullet\overline{C} + A\bullet C\bullet\overline{D}$$

- Egyszerűsítet függvény megvalósítása kapukkal

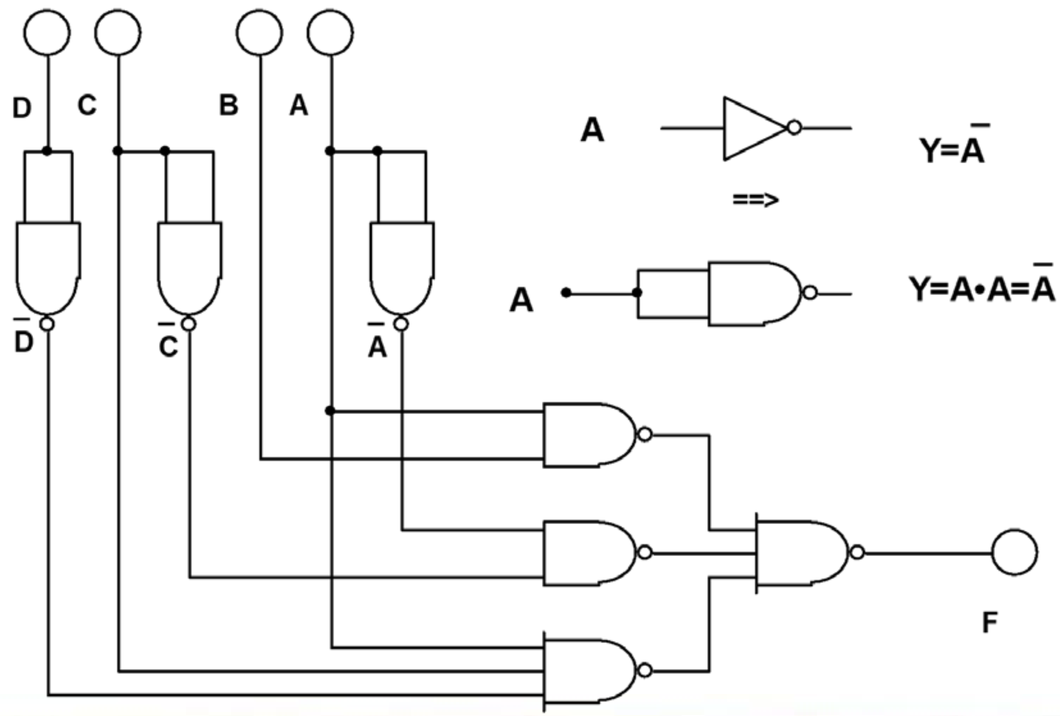


# A függvény megvalósítassa NAND kapukkal

$$F = A \bullet B + \overline{A} \bullet \overline{C} + A \bullet C \bullet \overline{D}$$

- Az előző függvény átalakítható De Morgan tételek használatával

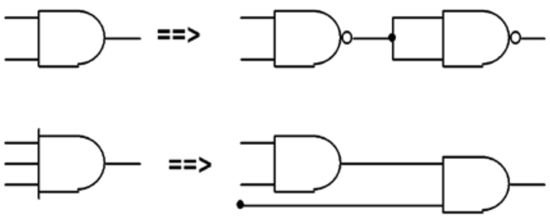
$$F = \overline{\overline{F}} = \overline{\overline{A \bullet B + \overline{A} \bullet \overline{C} + A \bullet C \bullet \overline{D}}} = \overline{\overline{A \bullet B} \bullet \overline{\overline{A} \bullet \overline{C}} \bullet \overline{A \bullet C \bullet \overline{D}}}$$



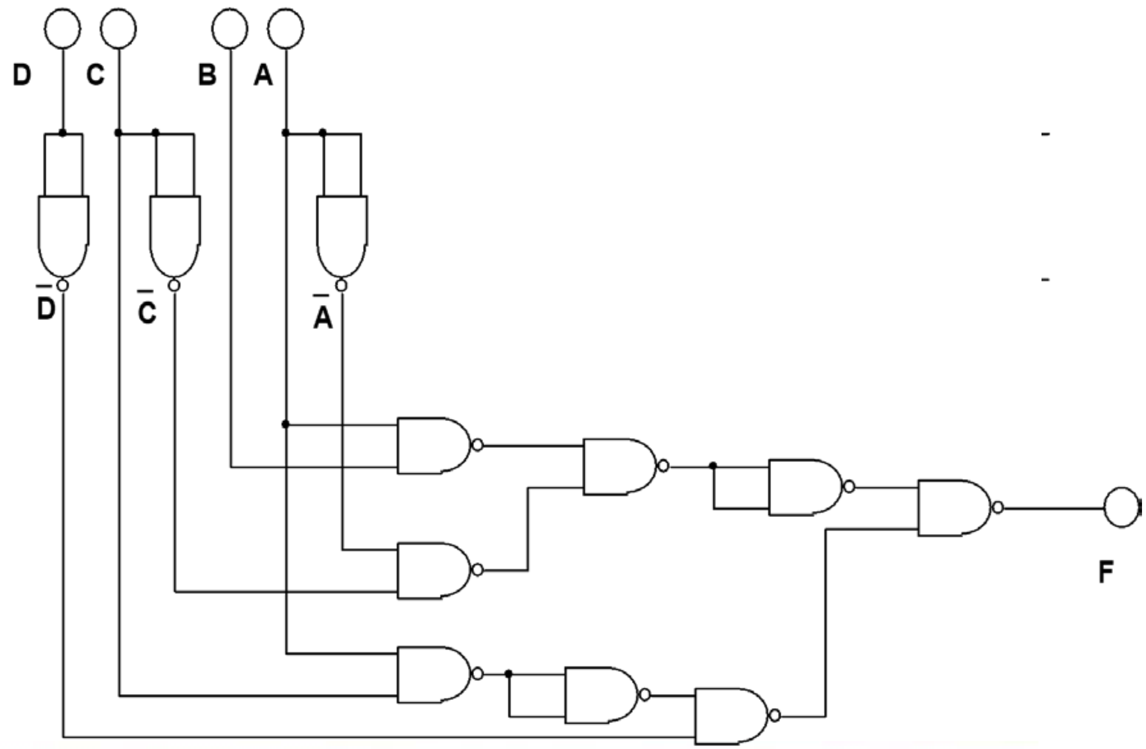
# A függvény megvalósítassa 2 bemenetű NAND kapukkal

- Kevesebb IC kapszula: egy adót típusú logikai kapu használata
- Az előző függvény átalakítható  az asszociativitás tulajdonság felhasználásával

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




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



# Logikai kapuk I

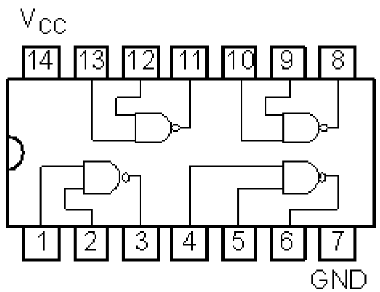
- Két fontosabb technológia:
  - TTL (Transistor-Transistor Logic)
  - CMOS (Complementary Metal-Oxide-Semiconductor).

- TTL kapuk
  - 5 V
  - előtag74 vagy 54
  - Fontosabb típusok:
    - 74 – normál TTL (betűk nélkül)
    - 74S – Schottky TTL
    - 74LS – Low-power Schottky TTL (kis fogyasztású TTL Schottky)
    - 74AS – Advanced Schottky TTL (nagy sebességű TTL Schottky)
    - 74ALS – Advanced Low-power Schottky TTL (nagy sebességű és kis fogyasztású TTL Schottky)
    - 74F – Fast TTL (gyors TTL)

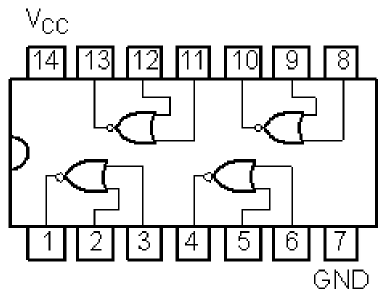
# Logikai kapuk II

- CMOS kapuk
  - 5V, 3,3V, 2,5V és 1,8V
  - előtag74 (vagy 54) + betű(k)
  - Fontosabb 5 V–os típusok:
    - −74HC és 74HCT – High-speed CMOS (T : TTL kompatibilis)
    - −74AC şi 74 ACT – Advanced CMOS (kis fogyasztású CMOS)
    - −74AHC şi 74AHCT - Advanced High-speed CMOS (nagy sebességű és kis fogyasztású CMOS)
  - −Fontosabb 3,3 V–os típusok:
    - −74LC – Low-voltage CMOS
    - −74LVC – Low-voltage CMOS
    - −74ALVC – Advanced Low-voltage CMOS

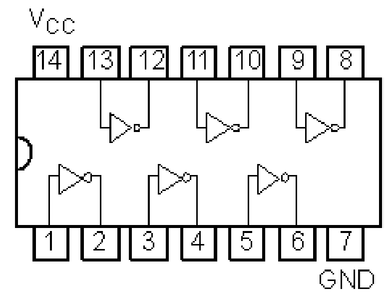
# Logikai kapuk III



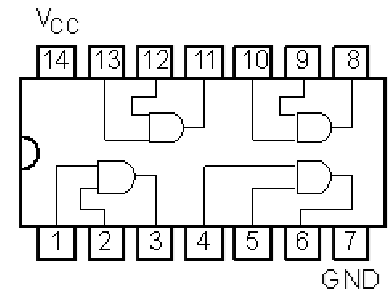
'00



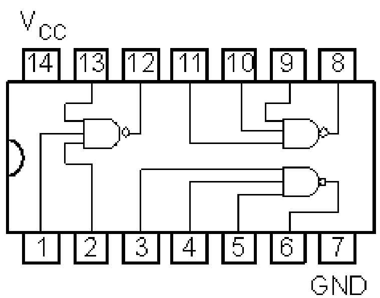
'02



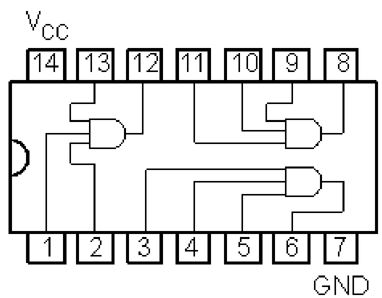
'04



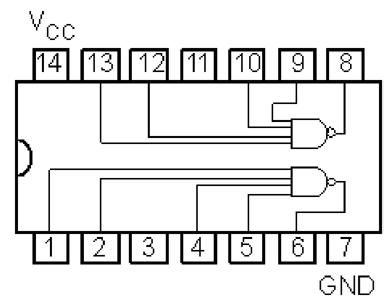
'08



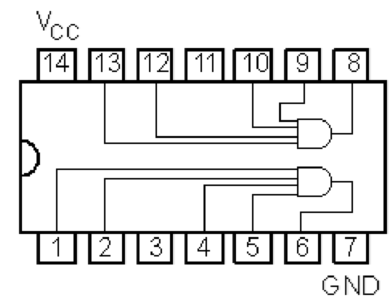
'10



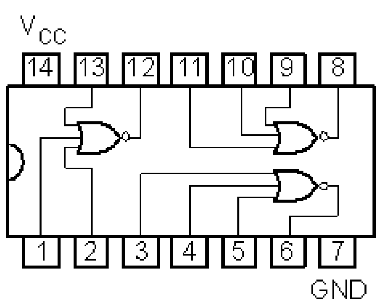
'11



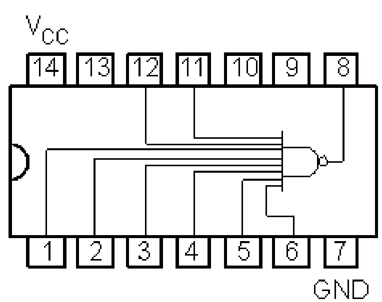
'20



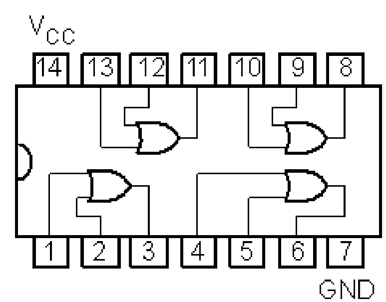
'21



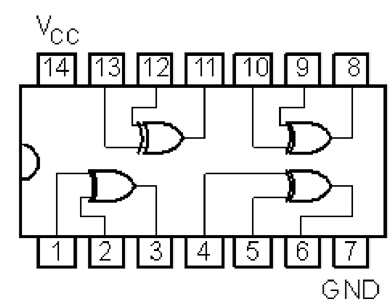
'27



'30



'32

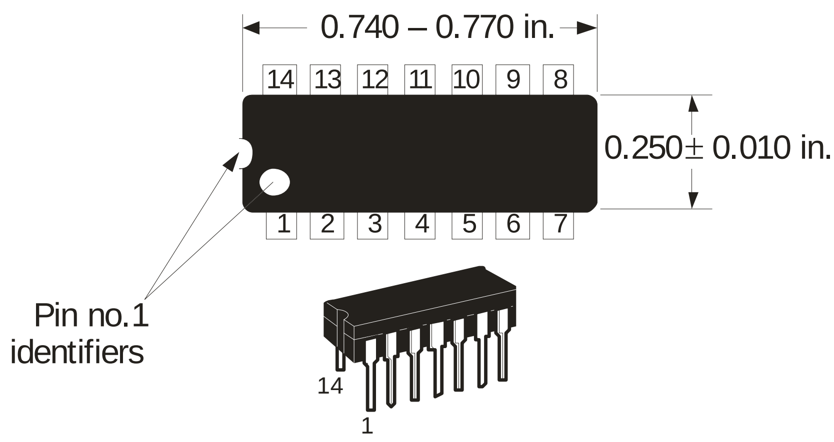


'86

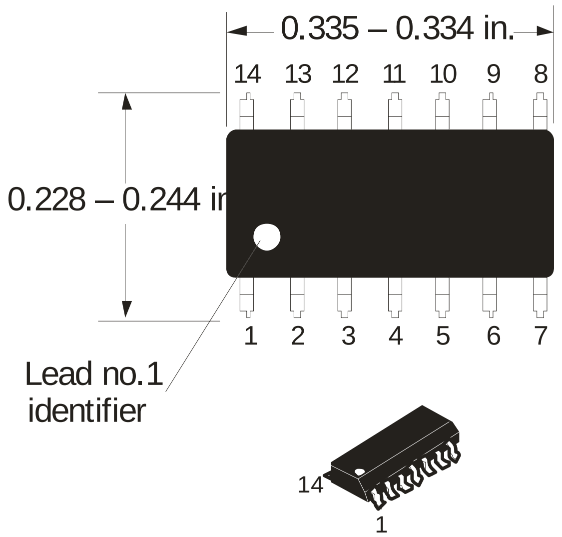
# Logikai kapuk III

Tokozás:

***DIP (Dual Inline Package)***
***SOIC (Small Outline Integrated Circuit)***



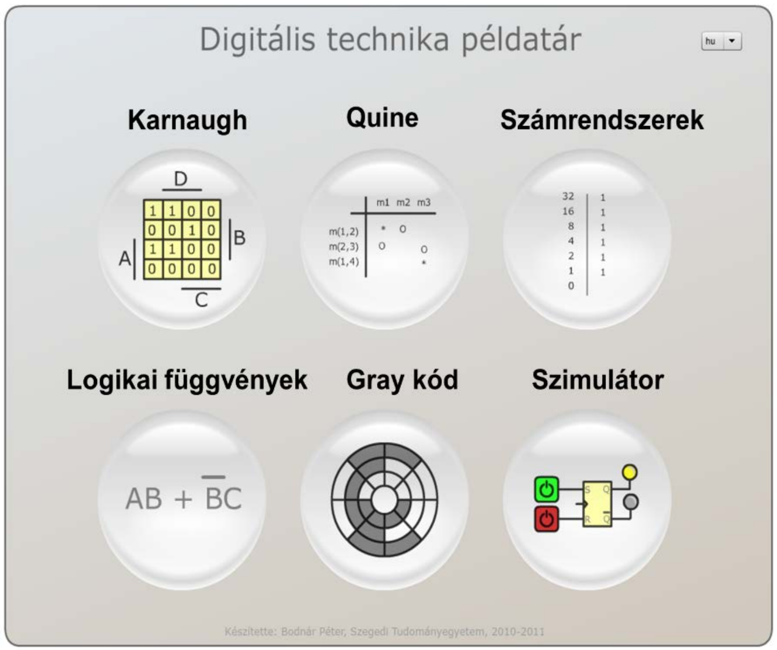

DIP package

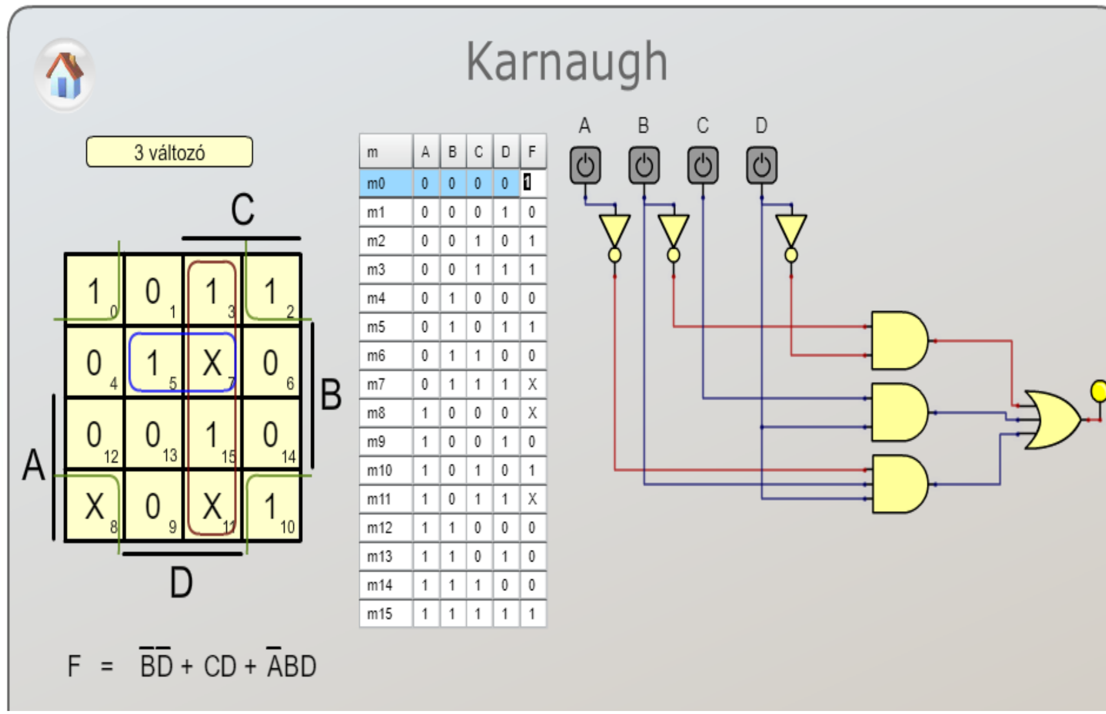



SOIC package

# Gyakorlat

- Digitális példatár – Link




- Online Karnaugh módszer – Link