BIK-TZP.21
Technologické základy počítačů

ZS 2021/22
2. sobota

doc. Ing. Kateřina Hyniová,CSc.
hyniova@fit.cvut.cz
Katedra číslicového návrhu, FIT ČVUT v Praze
kancelář A:1033

Přednáška 2C
Základní logická hradla

1. Základní logické členy

2. Booleova algebra

3. Základní logická hradla

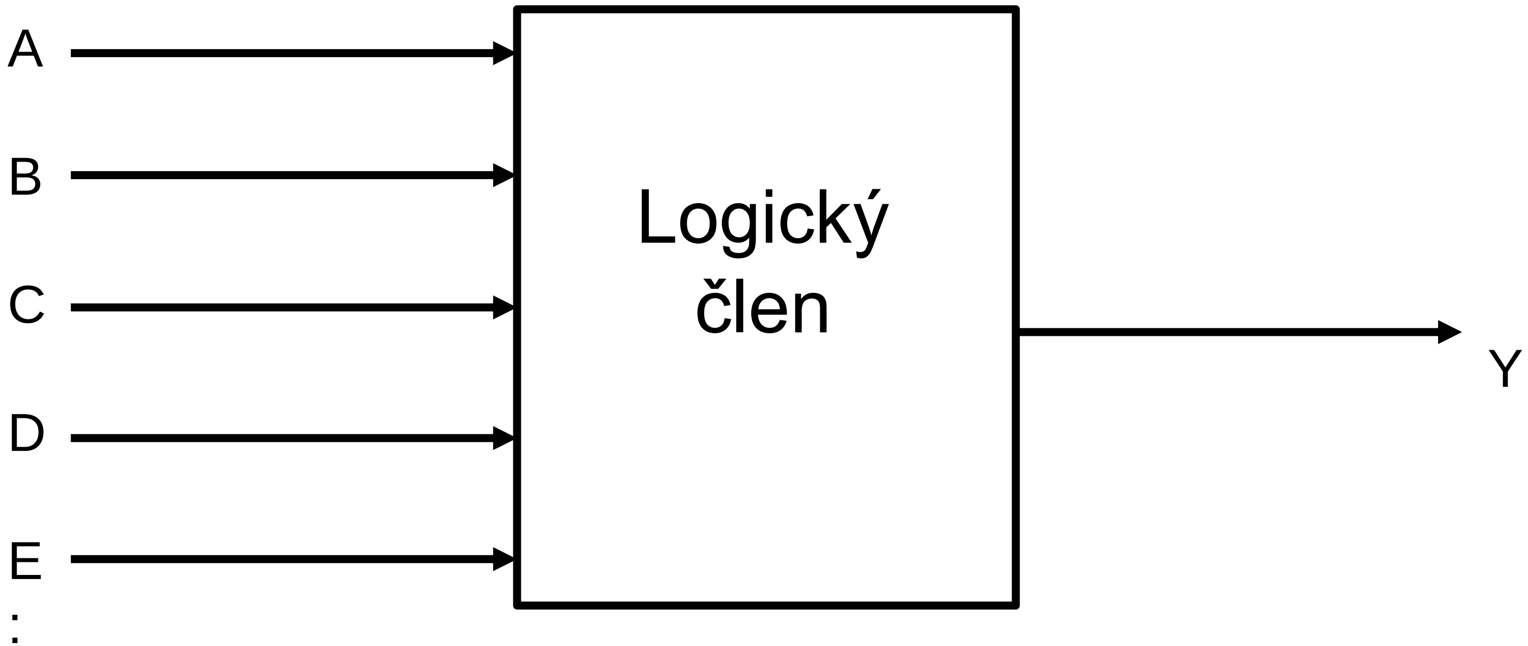
4. Booleova algebra

5. Návrh kombinačních logických obvodů -

- Příklady

1. Základní logické členy

- **Logický člen** neboli **hradlo** je základní stavební prvek logických obvodů, který vyčísluje logickou funkci.
- Typicky má jeden či více vstupů a jediný výstup.
- Hodnota na výstupu logického členu je funkcí logických hodnot 1,2 či více vstupních logických proměnných: Y=f(A,B,C,D,E…..)



Logický člen je obvod, kterým se realizují logické funkce. Logický signál, který tyto obvody zpracovávají, nabývá pouze 2 hodnot:

ANO	TRUE (pravda)	log 1
NE	FALSE (nepravda)	log 0

0 a 1 mají dvojí význam

-aritmetický – pro binární kódování čísel a pro aritmetické operace)

-logický – jako FALSE a TRUE (pro logické operace)

2. Booleova algebra

Booleova algebra je matematická disciplína, která je přímo aplikovatelná při návrhu číslicových obvodů. Tato 2-hodnotová algebra (0,1) zahrnuje pravidla a teorémy pro operace s logickými proměnnými a funkcemi. Při používání pravidel se využívají tři základní operace:

- **logický součin (konjunkce),**
- **logický součet (disjunkce),**
- **negace (inverze, doplněk),**

které tvoří teoretický prostředek pro návrh (syntézu) logických obvodů s požadovaným chováním.

2. Logické funkce

Hradlo	**Operace**	**Zápis**
NOT	Negace	$\overline{A}$
AND	Logický součin	$A \, . \, B$
OR	Logický součet	$A + B$
NAND	Negace logického součinu	$\overline{(A \, . \, B)}$
NOR	Negace logického součtu	$\overline{(A + B)}$
XOR	Non-ekvivalence	$A \oplus B$
XNOR	Ekvivalence	$\overline{(A \oplus B)}$

3. Základní logická hradla

- **Základní logické hradlo implementuje některou z logických operací**, ve schématech je tato logická operace reprezentována odpovídajícím symbolem hradla.
- Z Booleových operací na Booleových proměnných můžeme skládat složitější výrazy a realizovat je pomocí základních hradel. Implemantaci složitějších logických výrazů nazýváme:

  - hradlo

  - kombinační logický obvod

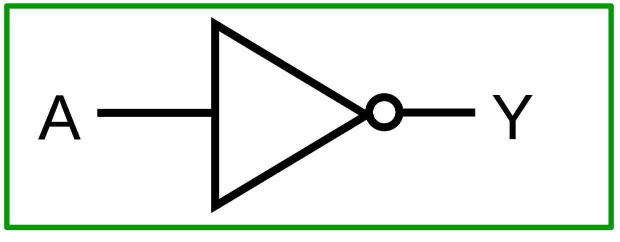
Booleovy funkce jedné proměnné

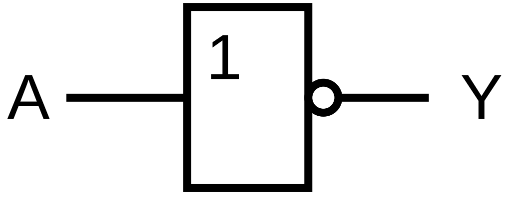
- **NOT** (negace, doplněk)
  - $Y = \neg A$
  - $Y = \overline{A}$
  - $Y = \text{not}(A)$
  - $Y = !A$

A	$Y = \overline{A}$
false	true
true	false

A	$Y = \overline{A}$
0	1
1	0

Logické hradlo (invertor)



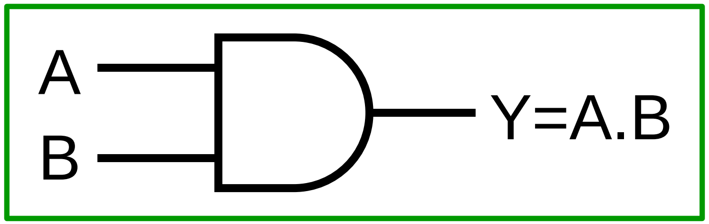


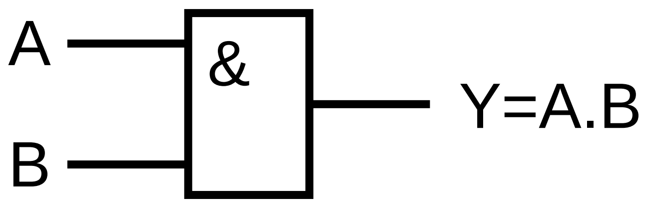
Booleovy funkce dvou proměnných

- **AND** (logický součin, konjunkce)
  - Y = A.B=AB
  - Y = A ∧ B
  - Y = A and B
  - Y = A & B

A	B	Y = A.B
false	false	false
false	true	false
True	false	false
True	true	true

Logické hradlo AND





A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

Booleovy funkce dvou proměnných

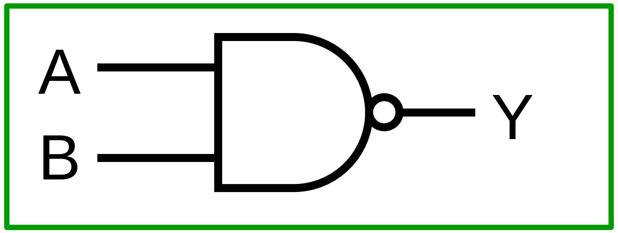
- **NAND**

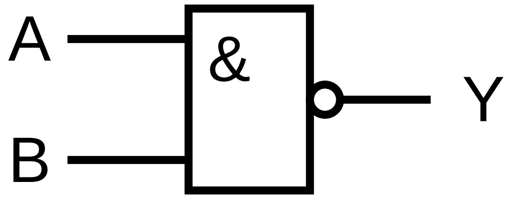
(negace logického součinu)

- $Y = \neg (A \wedge B)$
- $Y = \overline{A.B}$
- Y = not (A and B)

A	B	$Y = \overline{A.B}$
false	false	true
false	true	true
true	false	true
true	true	false

Logické hradlo NAND





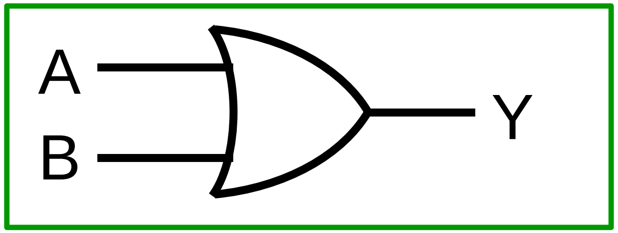
A	B	$Y = \overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

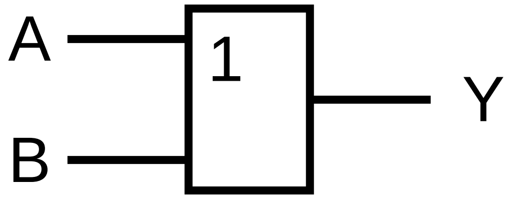
Booleovy funkce dvou proměnných

- **OR** (logický součet, disjunkce)
  - Y = A ∨ B
  - Y = A+B
  - Y = A or B
  - Y = A || B

A	B	Y = A+B
false	false	false
false	true	true
true	false	true
true	true	true

Logické hradlo OR





A	B	Y = A+B
0	0	0
0	1	1
1	0	1
1	1	1

Booleovy funkce dvou proměnných

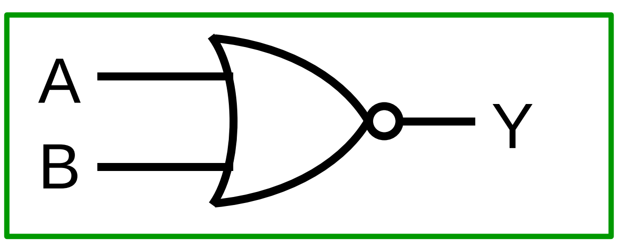
- **NOR**

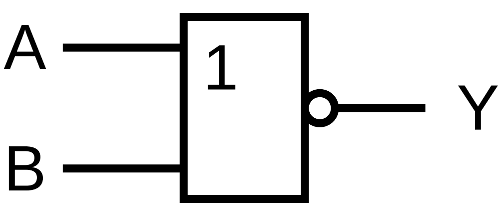
(Negace logického součtu)

  - $Y = \neg\,(A \vee B)$
  - $Y = \overline{A+B}$
  - Y = not (A or B)

A	B	$Y = \overline{(A+B)}$
false	false	true
false	true	false
true	false	false
true	true	false

Logické hradlo NOR





A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Booleovy funkce dvou proměnných

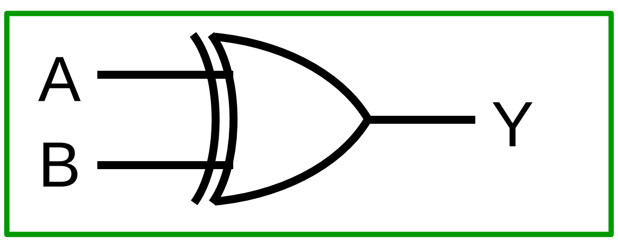
- **XOR**

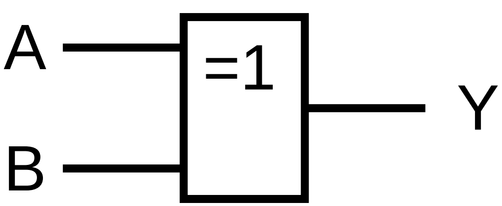
(exclusive or, non-ekvivalence)

  - $Y = A \not\Leftrightarrow B$
  - $Y = A \oplus B$
  - Y = A xor B

A	B	Y = A⊕B
false	false	false
false	true	true
true	false	true
true	true	false

Logické hradlo XOR





A	B	Y=A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Booleovy funkce dvou proměnných

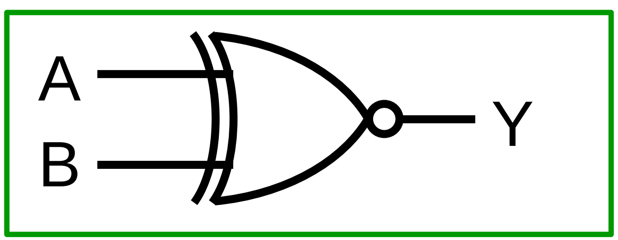
- **XNOR**

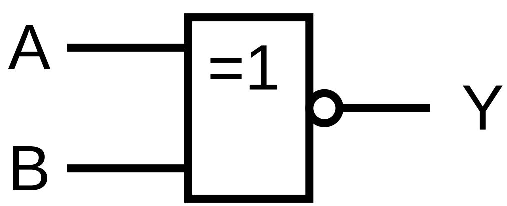
(exclusive nor, ekvivalence)

- $Y = A \Leftrightarrow B$
- $Y = \overline{A \oplus B}$
- Y = A xnor B

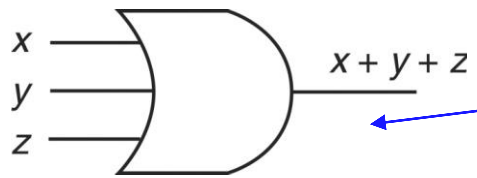
A	B	$Y = \overline{A \oplus B}$
false	false	true
false	true	false
true	false	false
true	true	true

Logické hradlo XNOR



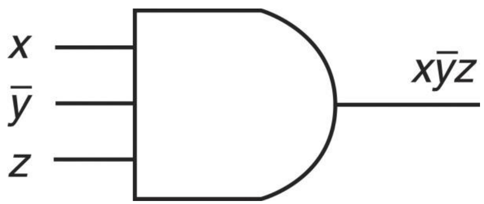


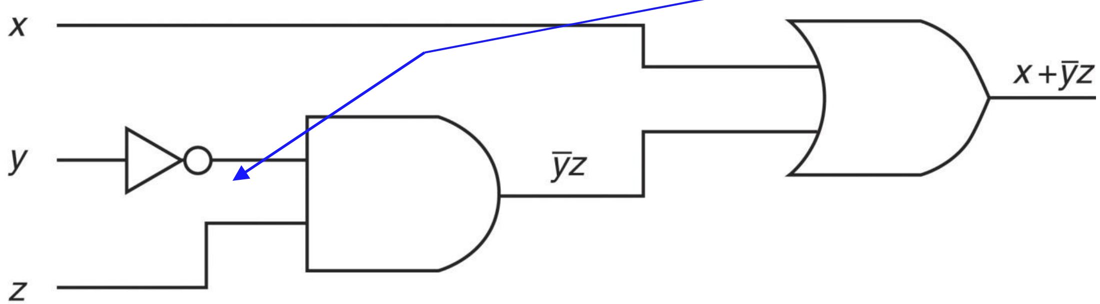
A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1



Hradla AND a OR mohou mít více než 2 vstupy

Pro negaci proměnné Y můžeme buď použít hradlo pro negaci (invertor) popř. jednoduše specifikovat invertovanou proměnnou $\overline{Y}$.

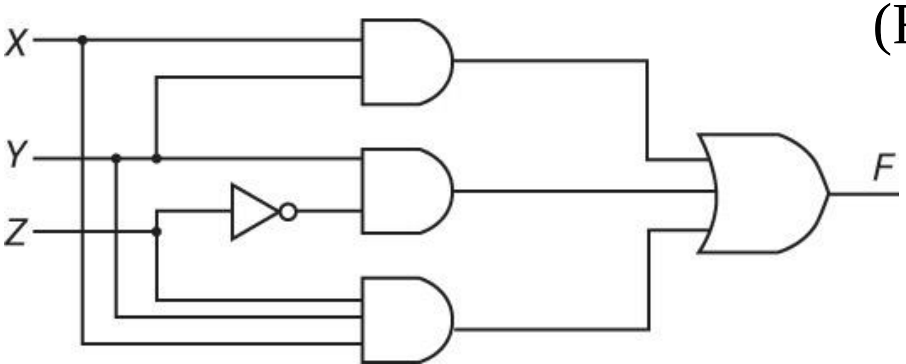




Pravdivostní tabulka obvodu

Inputs					Outputs
x	y	z	$\bar{y}$	$\bar{y}z$	$x+\bar{y}z = F$
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

- **Otázka:** Jaký logický výraz implementuje hradlo? (F=?)



4. Booleova algebra

- Booleova aritmetika
- Booleovy algebraické identity
- Booleovy algebraické vlastnosti
- De Morganovy zákony

Booleova aritmetika

- Vysvětluje součet a součin logických hodnot (vyplývá z pravdivostních tabulek)
- Logický součet koresponduje s normálně otevřenými spínači zapojenými paralelně
- Logický součin koresponduje s normálně otevřenými spínači zapojenými sériově

Booleova aritmetika–logický součet

**Logický součet: Součet libovolné logické hodnoty a 1 je vždy
roven 1, součet libovolné logické hodnoty a 0 je roven
původní logické hodnotě.**

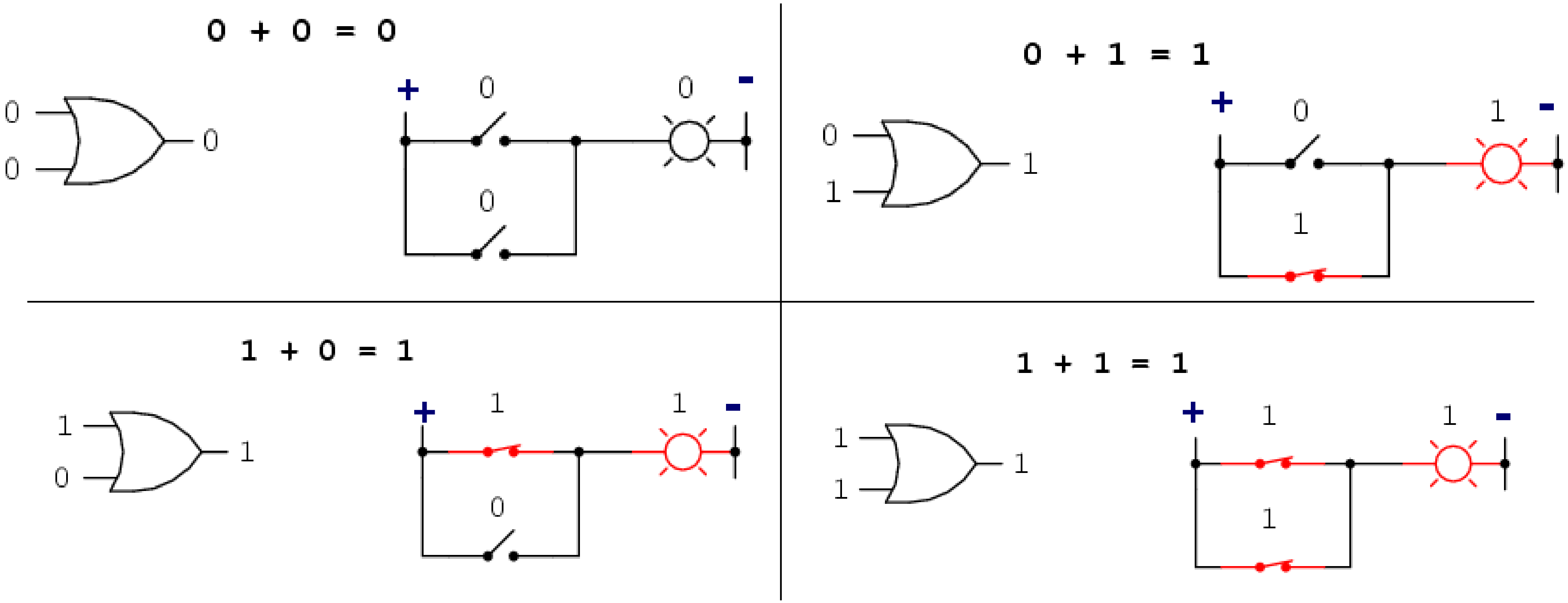
$$0 + 1 = 1$$

$$1 + 1 = 1$$

$$0 + 0 = 0$$

$$1 + 0 = 1$$

Logický součet OR reprezentovaný N.O. spínači zapojenými paralelně



N.O. spínač – normálně otevřený spínač

Booleova aritmetika–logický součin

Logický součin : Libovolná logická hodnota násobená 0 je rovna 0, libovolná logická hodnota násobená 1 se nemění .

$$0 \;.\; 0 = 0$$

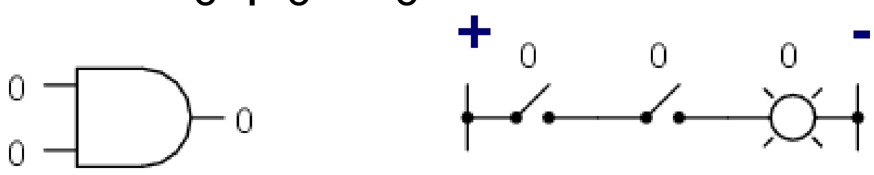
$$1 \;.\; 0 = 0$$

$$0 \;.\; 1 = 0$$

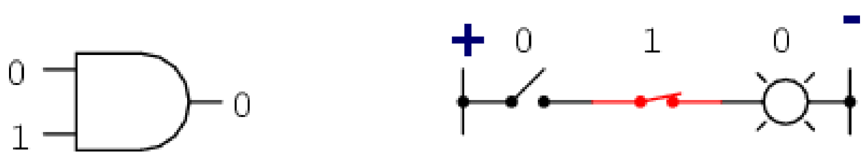
$$1 \;.\; 1 = 1$$

Logický součin AND reprezentovaný N.O. spínači zapojenými sériově

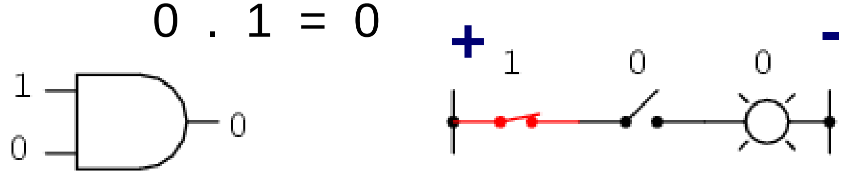
$0 \cdot 0 = 0$



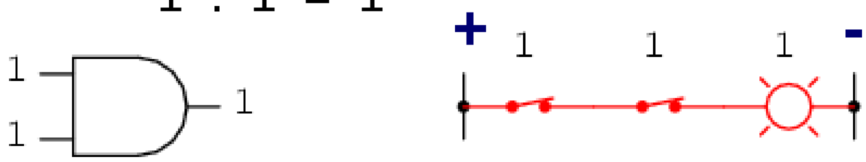
$1 \cdot 0 = 0$



$0 \cdot 1 = 0$

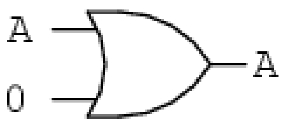


$1 \cdot 1 = 1$

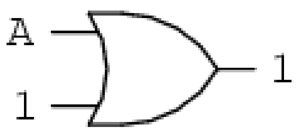


Booleovské algebraické identity

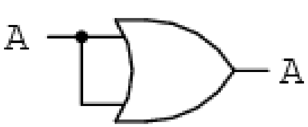
$A + 0 = A$



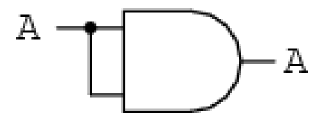
$A + 1 = 1$



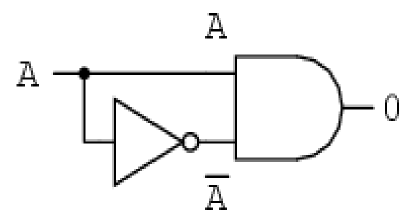
$A + A = A$

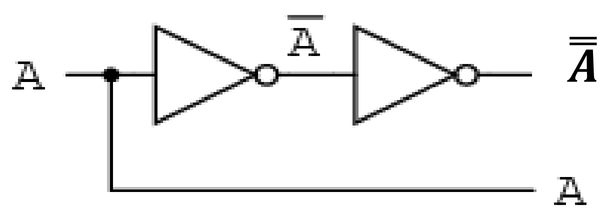


$AA = A$



$A\bar{A} = 0$





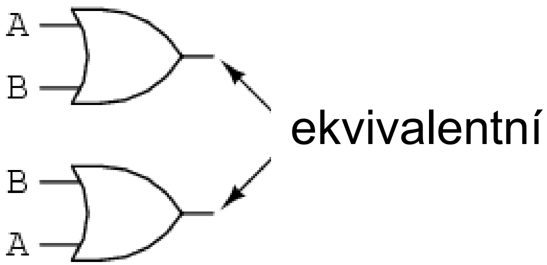
$\bar{\bar{A}} = A$

Booleovy algebraické vlastnosti

komutativní

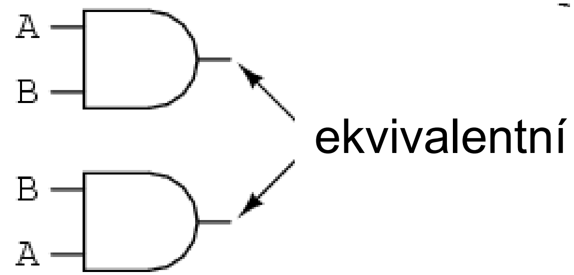
Logický součet

$$A + B = B + A$$



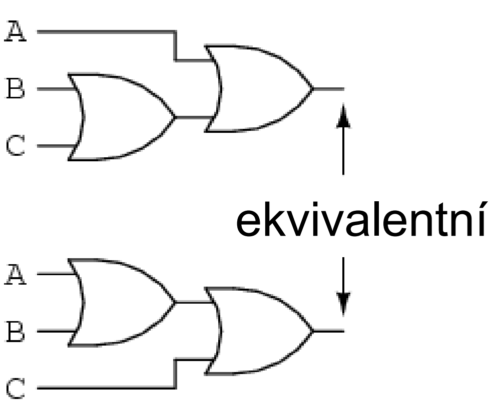
Logický součin

$$AB = BA$$

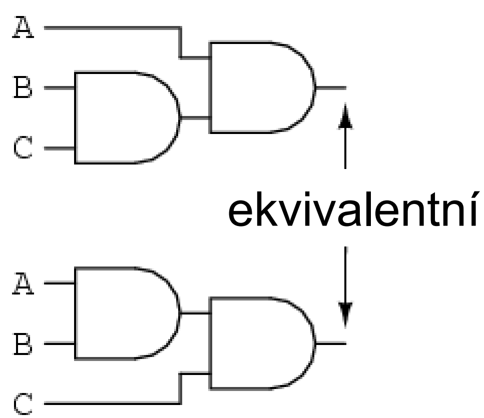


asociativní

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



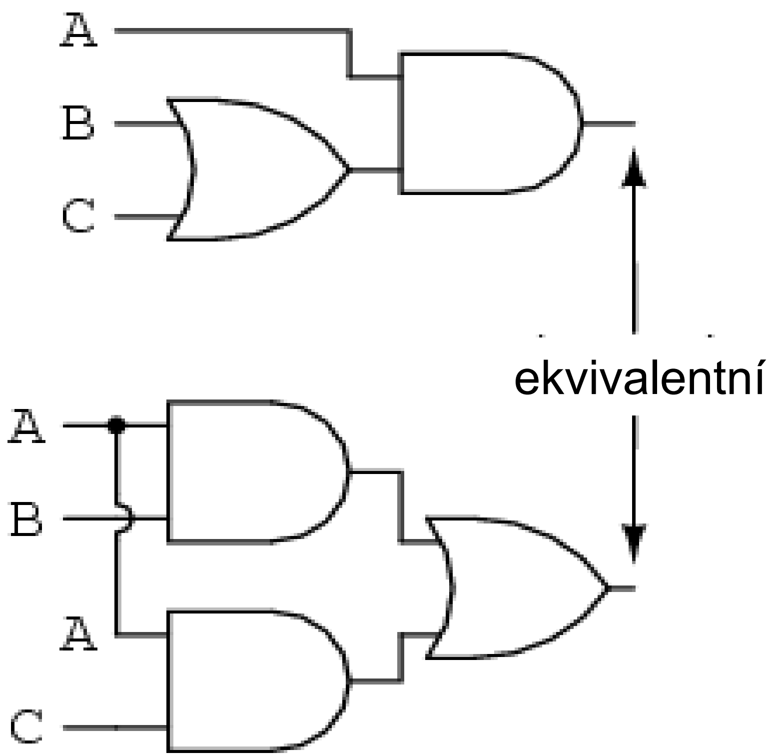
$$A(BC) = (AB)C$$



Booleovy algebraické vlastnosti

distributivní

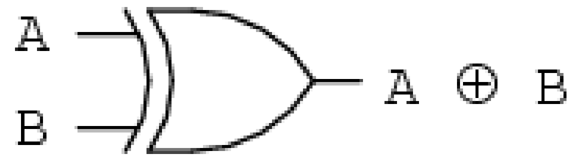
$$A(B + C) = AB + AC$$



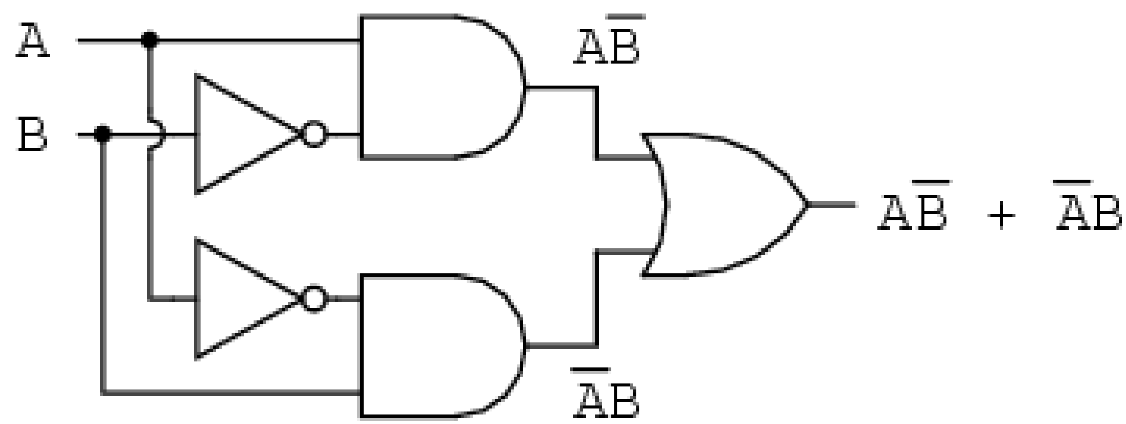
Funkce Exclusive-OR

$$A \oplus B = A\overline{B} + \overline{A}B$$

A	B	Y = A⊕B
0	0	0
0	1	1
1	0	1
1	1	0



….je ekvivalentní zapojení….



$$A \oplus B = A\overline{B} + \overline{A}B$$

De Morganovy zákony

De Morganovy zákony

1) $\overline{X + Y} = \bar{X} . \bar{Y}$

2) $\overline{X . Y} = \bar{X} + \bar{Y}$

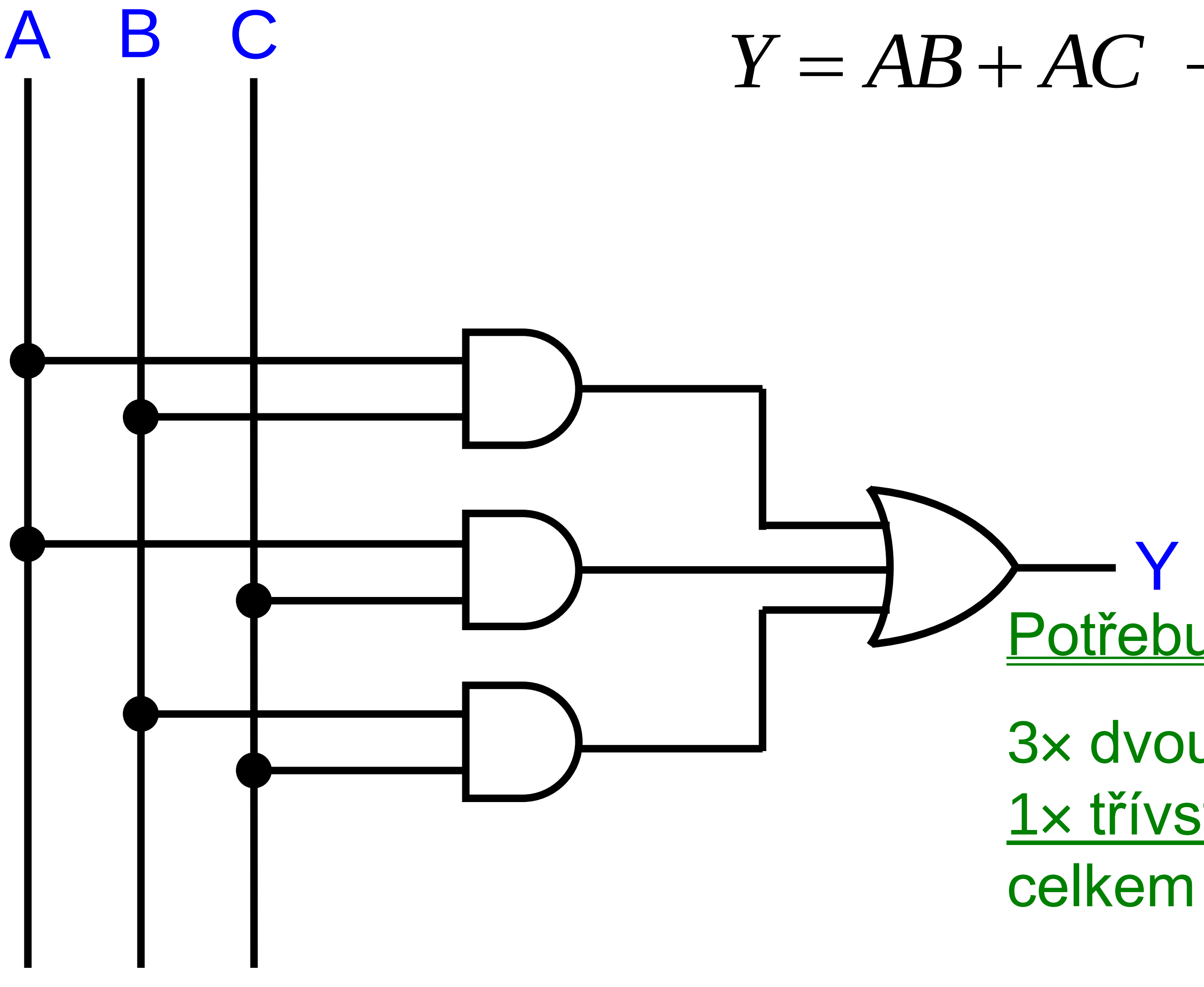
Zákony tedy říkají:

- **1) Negace logického součtu libovolného počtu proměnných je rovna součinu negovaných těchto proměnných.**

- **2) Negace logického součinu libovolného počtu proměnných je rovna součtu negovaných těchto proměnných.**

Příklad #1:

$$Y = AB + AC + BC$$



Potřebujeme:

3× dvouvstupový AND
1× třívstupový OR
celkem 26 tranzistorů)

Použijeme De Morganovy zákony:

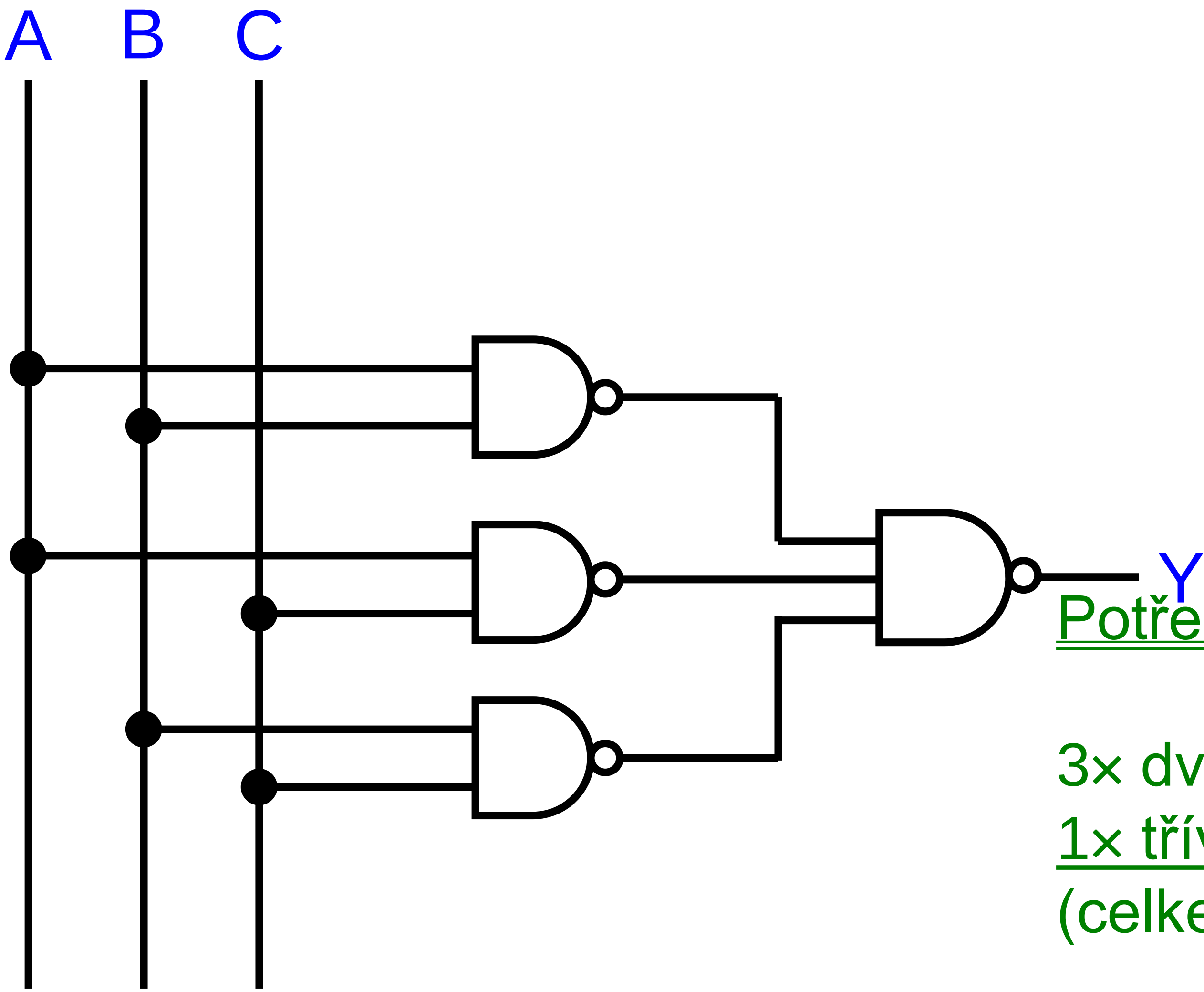
$$Y = AB + AC + BC$$

$$Y = \overline{\overline{AB + AC + BC}}$$

$$Y = \overline{\overline{AB} \cdot \overline{AC} \cdot \overline{BC}}$$

$$\boxed{\begin{aligned} \overline{X + Y} &= \bar{X} . \bar{Y} \\ \overline{X . Y} &= \bar{X} + \bar{Y} \end{aligned}}$$

$$Y = AB + AC + BC = \overline{\overline{AB} \cdot \overline{AC} \cdot \overline{BC}}$$



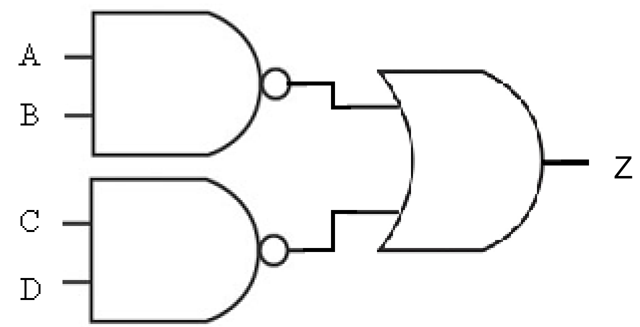
Potřebujeme:

3× dvouvstupový NAND
1× třívstupový NAND
(celkem 18 tranzistorů)

5. Návrh kombinačních obvodů - Příklady

Návrh kombinačních log. obvodů

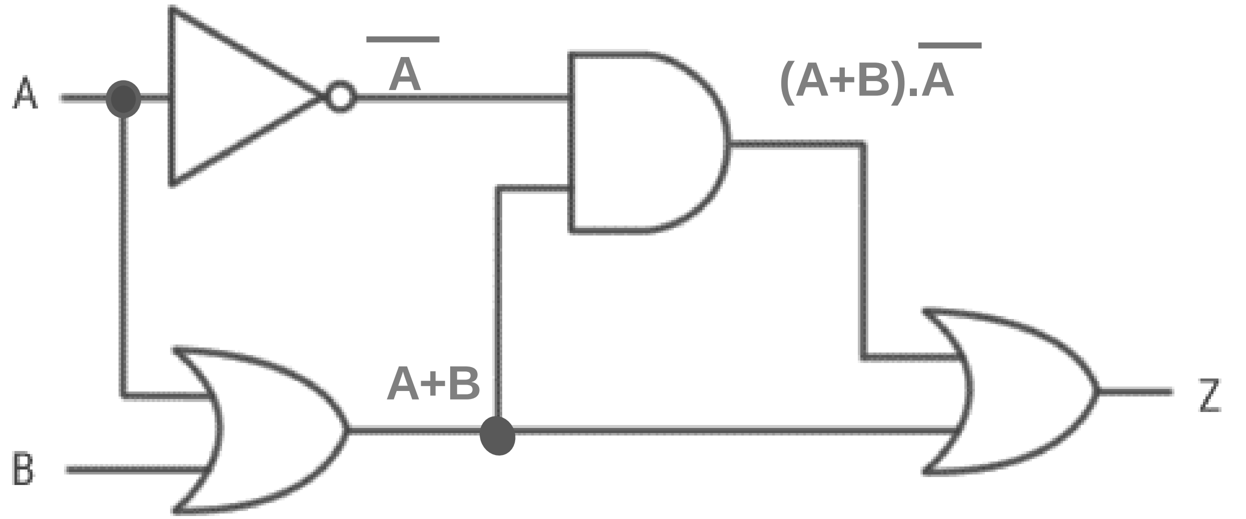
- Kombinováním základních logických hradel můžeme realizovat složitější logické výrazy.
- Např.: $Z = \overline{A.B} + \overline{C.D}$
- Výsledný obvod je kombinační logický obvod



- Takové řešení nemusí být optimální co do počtu použitých hradel i struktury obvodu. Optimalizovat se naučíte v BIK-SAP v LS.
- Takto jsou realizovány číslicové obvody, které tvoří ALU a jiné části počítačů.

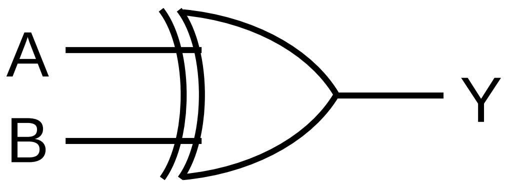
Příklad #2: Implementujte logickou funkci Z(A,B) pomocí základních log. hradel):

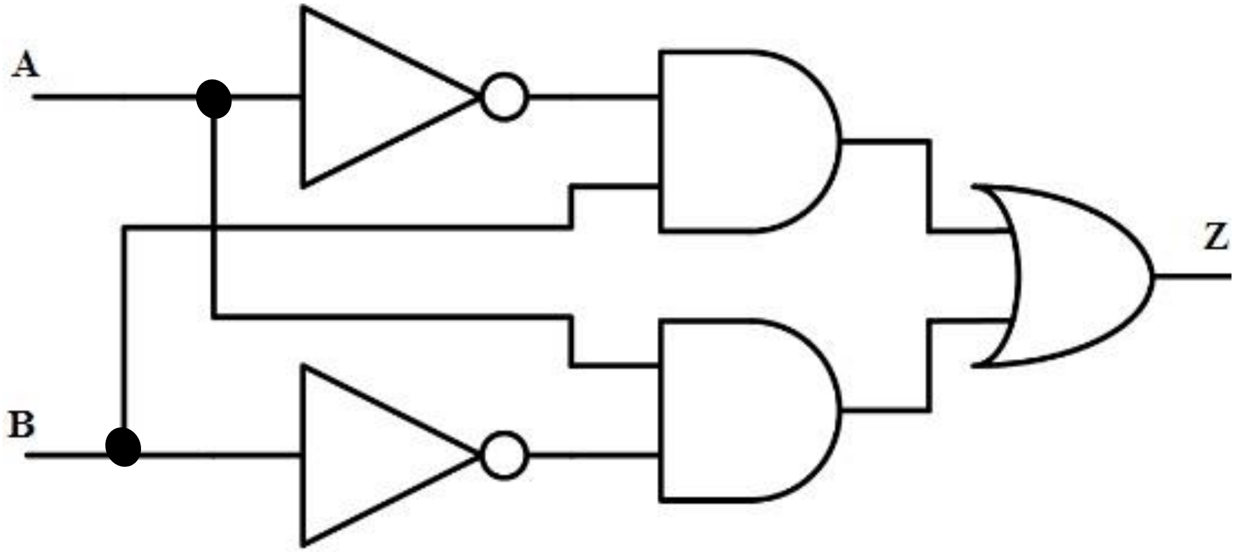
$$Z(A,B)=(A + B).\overline{A} + (A + B)$$



<u>Příklad #3</u>: Implementujte logickou funkci Z(A,B) pomocí základních log. hradel):

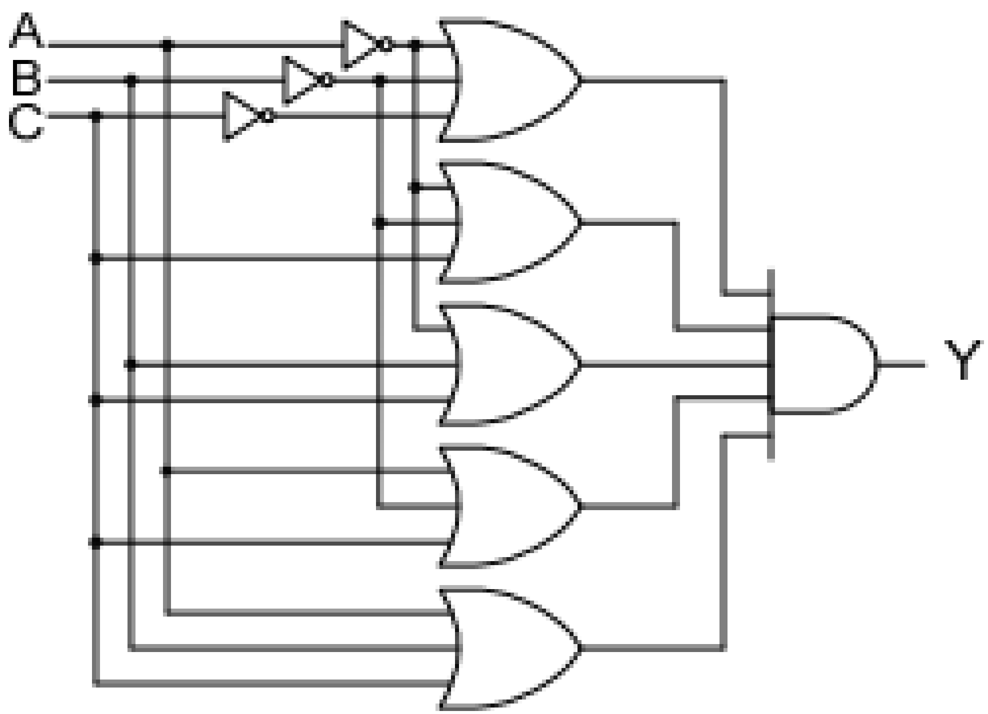
$$Z(A,B)=\overline{A}.B + A.\overline{B} = A \oplus B$$





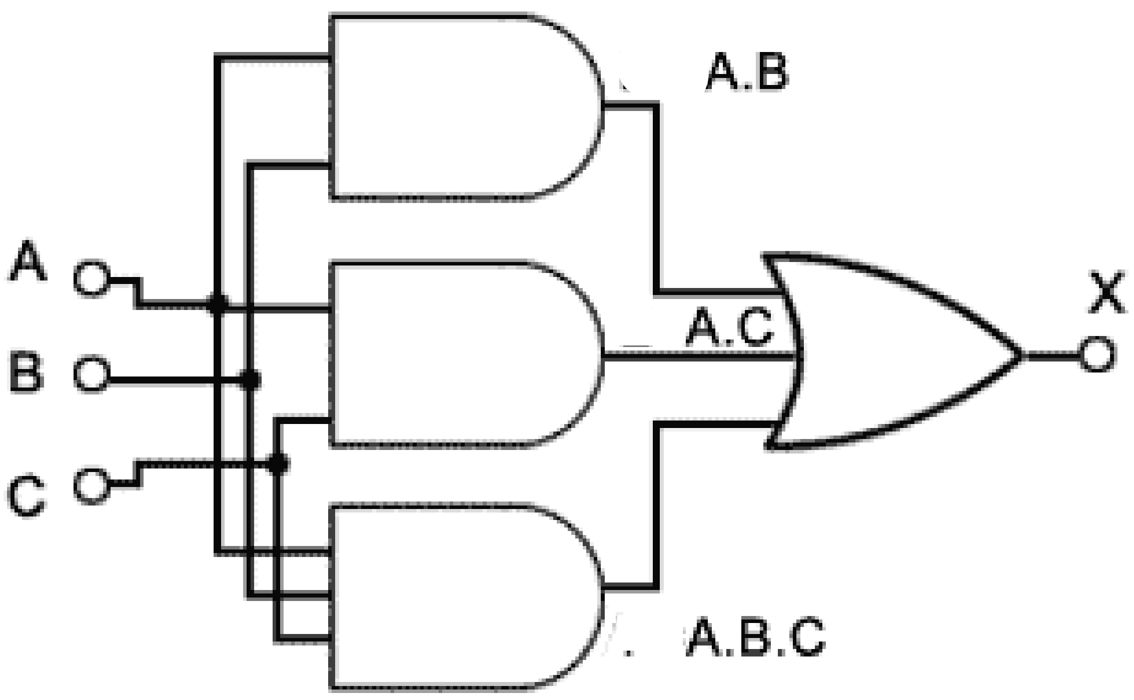
<u>Příklad #4</u>: Implementujte logickou funkci Y(A,B,C) pomocí základních log. hradel):

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



Příklad #5: Implementujte logickou funkci X(A,B,C) pomocí základních log. hradel):

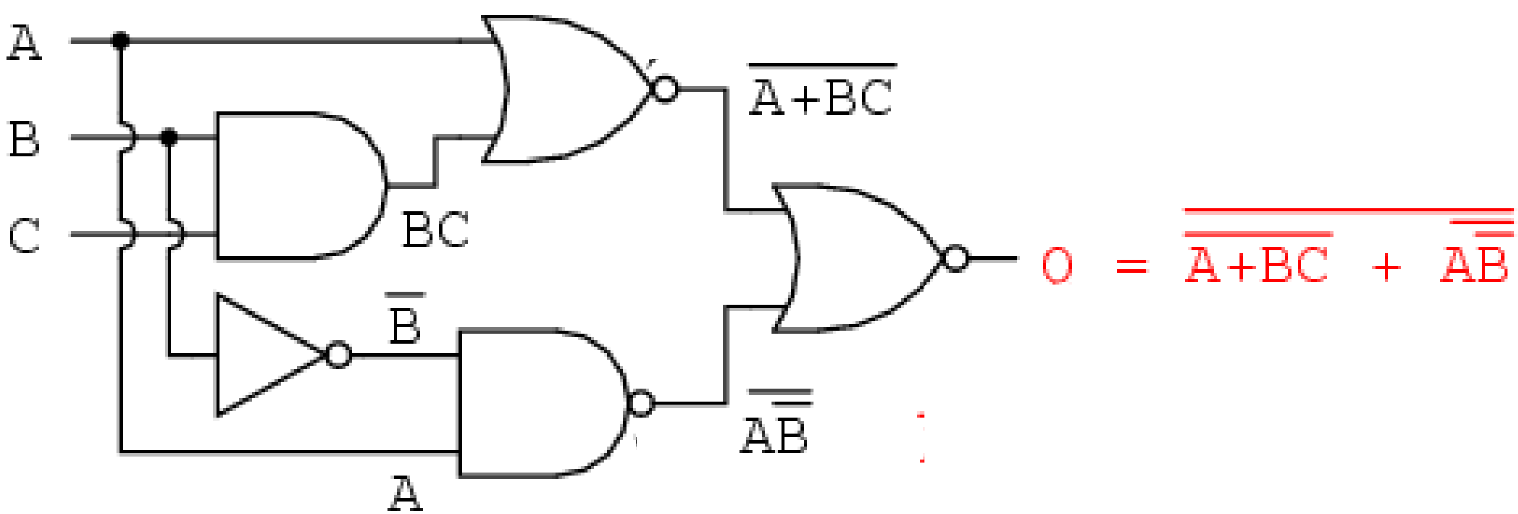
$$X(A,B,C) = (A.B)+(A.C)+(A.B.C)$$



X= (A.B)+(A.C)+(A.B.C)

<u>Příklad #6</u>: Implementujte logickou funkci Q(A,B,C) pomocí základních log. hradel):

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



Příklad #7: implementujte log. funkci Q(A,B,C) 2–vstupovými základ. logickými hradly:

$$Q(A,B,C) = AB+B.C.(B+C)$$