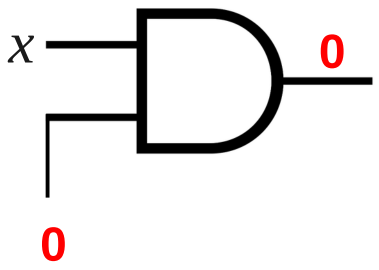
# Boolean Algebra (A1)

Young Won Lim
5/4/25

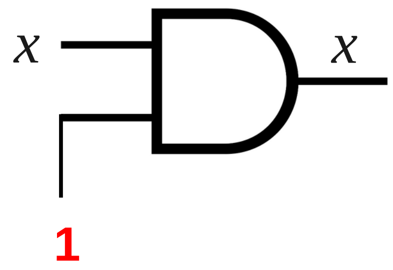
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

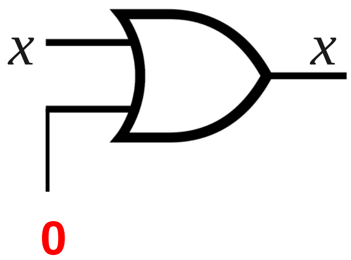
# Identity and Null Element Theorem



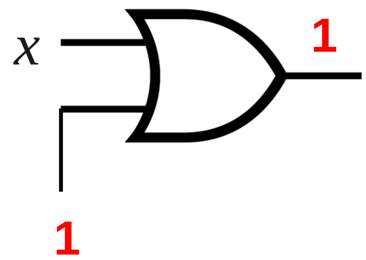

$$x \cdot 0 = 0$$



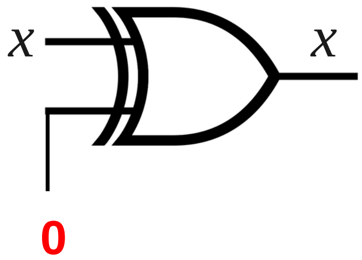

$$\boxed{x \cdot 1 = x}$$



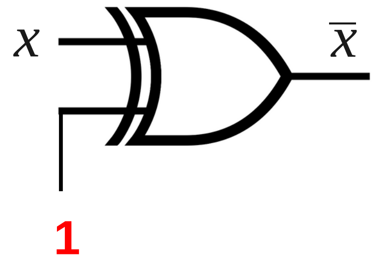

$$\boxed{x + 0 = x}$$




$$x + 1 = 1$$




$$\boxed{x \oplus 0 = x}$$




$$x \oplus 1 = \bar{x}$$

# Distributive

$$\underbrace{x \cdot (y + z)}_{} = x \cdot y + x \cdot z \neq x \cdot y + z$$

This parenthesis cannot be deleted

$$x + (y \cdot z) = (x + y) \cdot (x + z) = x + y \cdot z$$

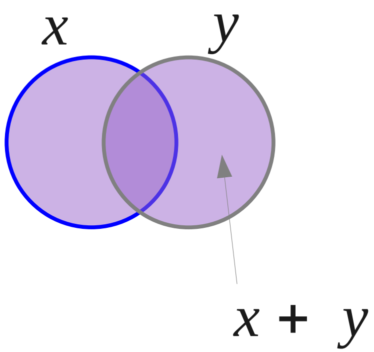
This parenthesis can be deleted

Operator precedence : $\cdot$ > $+$

# Inclusion

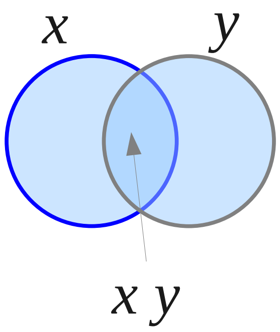
$$x \cdot (x + y) = x$$

$$\begin{aligned} x \cdot (x + y) &= x \cdot x + x \cdot y \\ &= x + x \cdot y \\ &= x \cdot (1 + y) \\ &= x \end{aligned}$$




$$x + xy = x$$

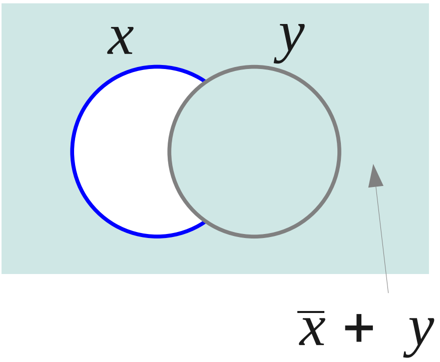
$$\begin{aligned} x + xy &= x \cdot 1 + x \cdot y \\ &= x \cdot (1 + y) \\ &= x \end{aligned}$$



# Eliminate

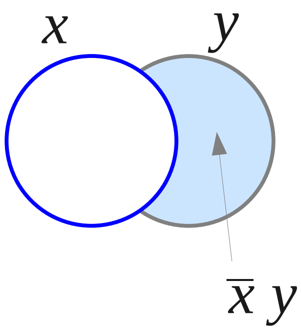
$$x \cdot (\overline{x} + y) = x\,y$$

$$x \cdot (\overline{x} + y) = x \cdot \overline{x} + x \cdot y$$
$$= 0 + x \cdot y$$
$$= x \cdot y$$




$$x + \overline{x}\,y = x + y$$

$$x + \overline{x}\,y = (x + \overline{x}) \cdot (x + y)$$
$$= 1 \cdot (x + y)$$
$$= x + y$$

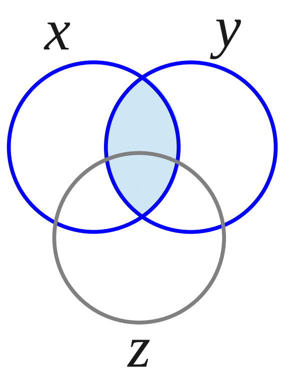


# Consensus (1)

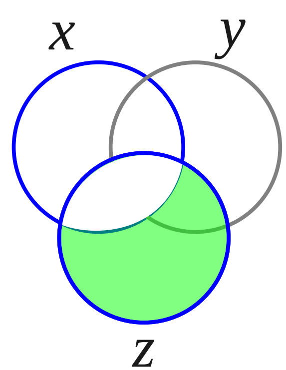
$$x \cdot y + \overline{x} \cdot z + \boxed{y \cdot z} = x \cdot y + \overline{x} \cdot z$$

redundant

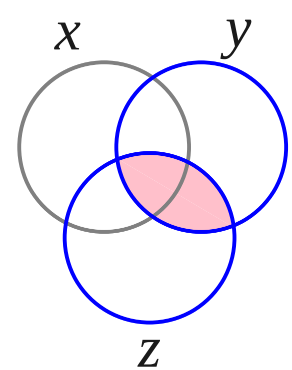
$x \cdot y$




$\overline{x} \cdot z$

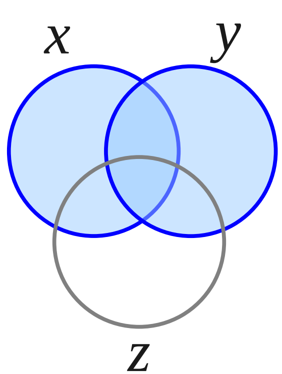



$y \cdot z$

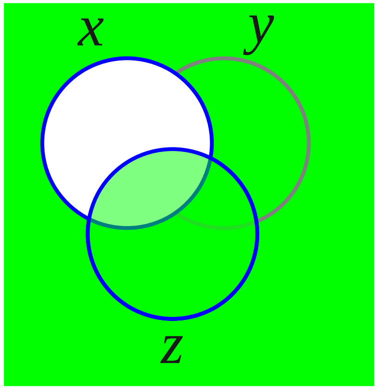



$$(x + y) \cdot (\overline{x} + z) \cdot (\boxed{y + z}) = (x + y) \cdot (\overline{x} + z)$$

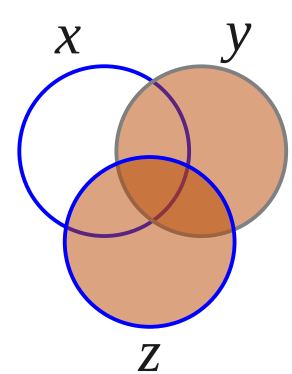
$x + y$




$\overline{x} + z$



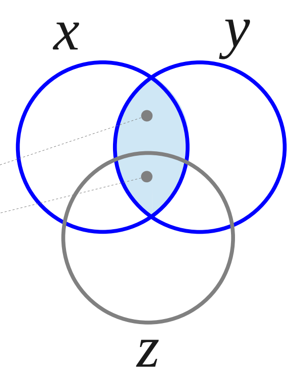

$y \cdot z$



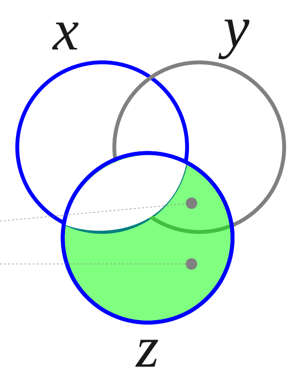
# Consensus (2)

$$x \cdot y + \overline{x} \cdot z + \boxed{y \cdot z} = x \cdot y + \overline{x} \cdot z$$

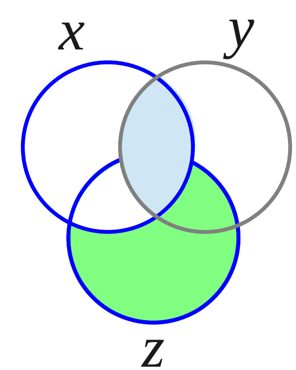
$x \cdot y =$
$x \cdot y \cdot \overline{z} +$
$x \cdot y \cdot z$




$\overline{x} \cdot z =$
$\overline{x} \cdot y \cdot z +$
$\overline{x} \cdot \overline{y} \cdot z$

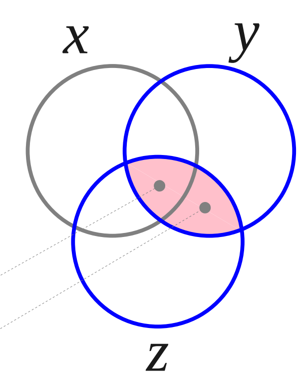



$x \cdot y +$
$\overline{x} \cdot z$




$$\begin{aligned} x \cdot y + \overline{x} \cdot z + y \cdot z &= x \cdot y + \overline{x} \cdot z + (x + \overline{x}) \cdot y \cdot z \\ &= x \cdot y + \overline{x} \cdot z + x \cdot y \cdot z + \overline{x} \cdot y \cdot z \\ &= x \cdot y \cdot (1 + z) + \overline{x} \cdot z \cdot (1 + y) \\ &= x \cdot y + \overline{x} \cdot z \end{aligned}$$

$y \cdot z =$
$x \cdot y \cdot z +$
$\overline{x} \cdot y \cdot z$




redundant

# Consensus (3)

$$x \cdot y + \overline{x} \cdot z + \boxed{y \cdot z} = x \cdot y + \overline{x} \cdot z$$

$x \cdot y$

| | $\overline{x} \cdot \overline{y}$ | $\overline{x} \cdot y$ | $x \cdot y$ | $x \cdot \overline{y}$ |
|---|---|---|---|---|
| $\overline{z}$ | | | 1 | |
| $z$ | 1 | 1 | 1 | |

$\overline{x} \cdot z$

| | $\overline{x} \cdot \overline{y}$ | $\overline{x} \cdot y$ | $x \cdot y$ | $x \cdot \overline{y}$ |
|---|---|---|---|---|
| $\overline{z}$ | | | 1 | |
| $z$ | 1 | 1 | 1 | |

$y \cdot z$

| | $\overline{x} \cdot \overline{y}$ | $\overline{x} \cdot y$ | $x \cdot y$ | $x \cdot \overline{y}$ |
|---|---|---|---|---|
| $\overline{z}$ | | | 1 | |
| $z$ | 1 | 1 | 1 | |

# (x+y)(x+z) = x+yz

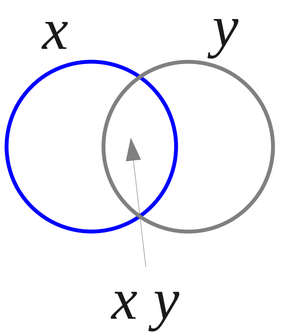
$$(x + y)(x + z) = x + yz$$

$$(x + y)(x + z) = xx + xz + xy + yz$$

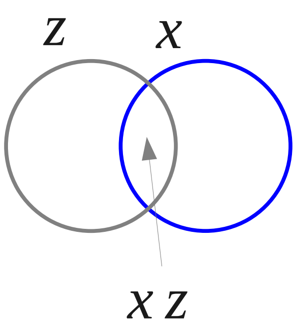
$$= \underline{x + xy} + xz + yz$$

$$= \underline{x + xz} + yz$$

$$= x + yz$$




$$x + xy = x$$




$$x + xz = x$$

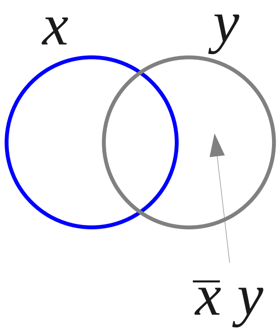
# x + x'y = x + y

$$x + \overline{x}\,y = x + y$$

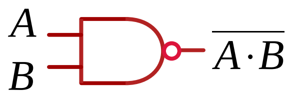
$$x + \overline{x}\,y = (x + \overline{x})\cdot(x + y)$$

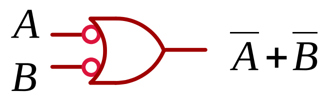
$$= 1 \cdot (x + y)$$

$$= x + y$$



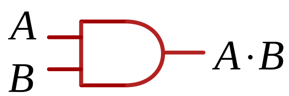
# De Morgan's Law (1)

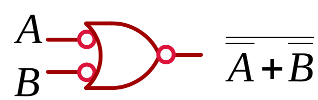



| A | B | $\overline{A \cdot B}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| $\overline{A}$ | $\overline{B}$ | $\overline{A}+\overline{B}$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

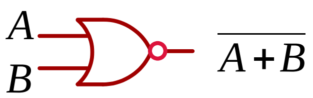


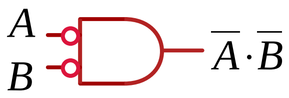




| A | B | $A \cdot B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| $\overline{A}$ | $\overline{B}$ | $\overline{\overline{A}+\overline{B}}$ |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

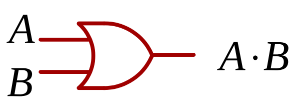
# De Morgan’s Law (2)

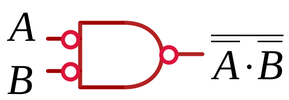



| A | B | $\overline{A+B}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

| $\overline{A}$ | $\overline{B}$ | $\overline{A}\cdot\overline{B}$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

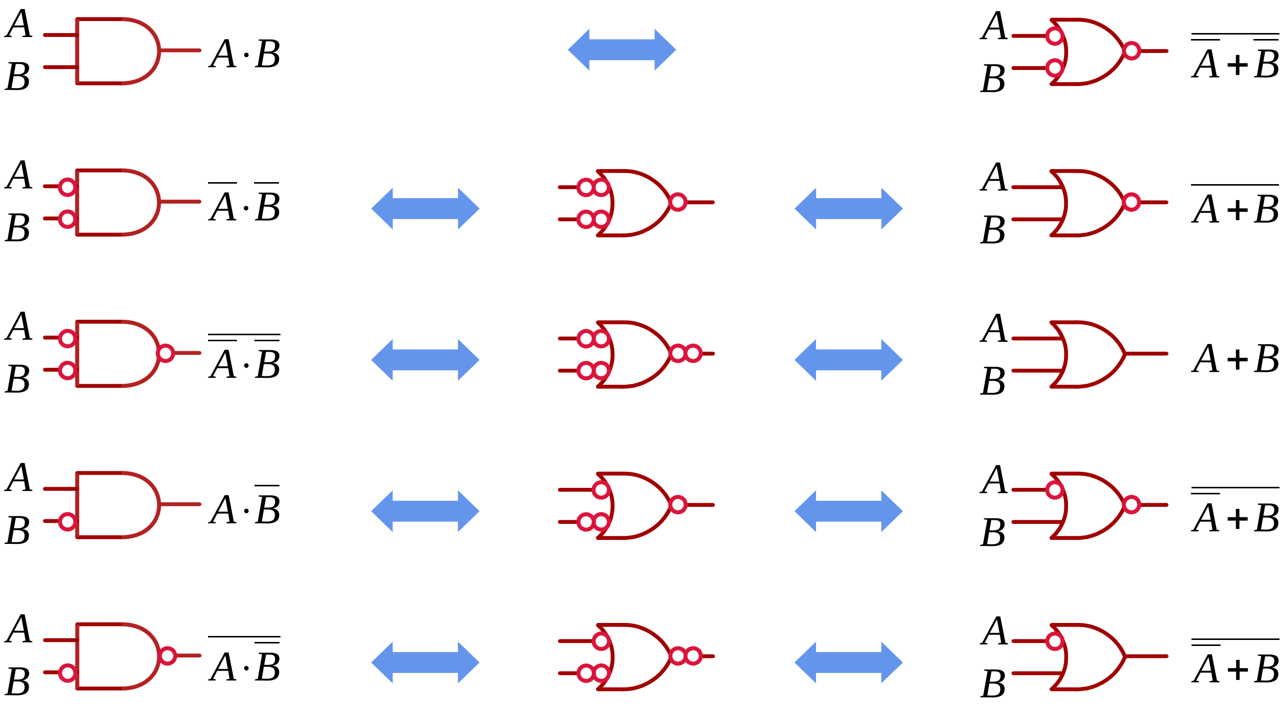


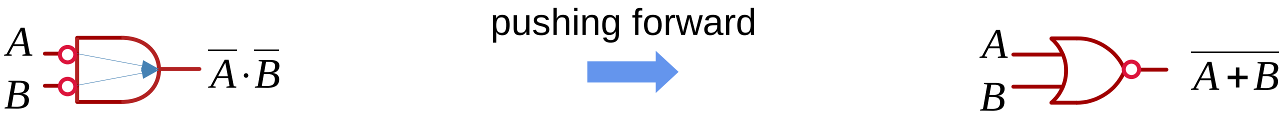

| A | B | $A\cdot B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

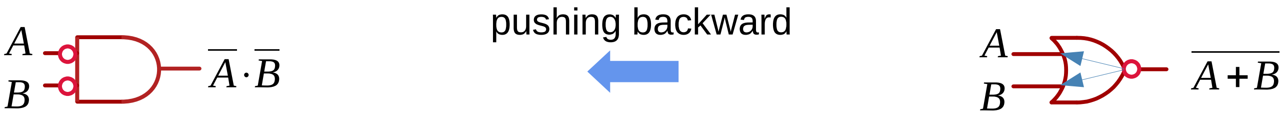
| $\overline{A}$ | $\overline{B}$ | $\overline{\overline{A}+\overline{B}}$ |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

# De Morgan’s Law (3)

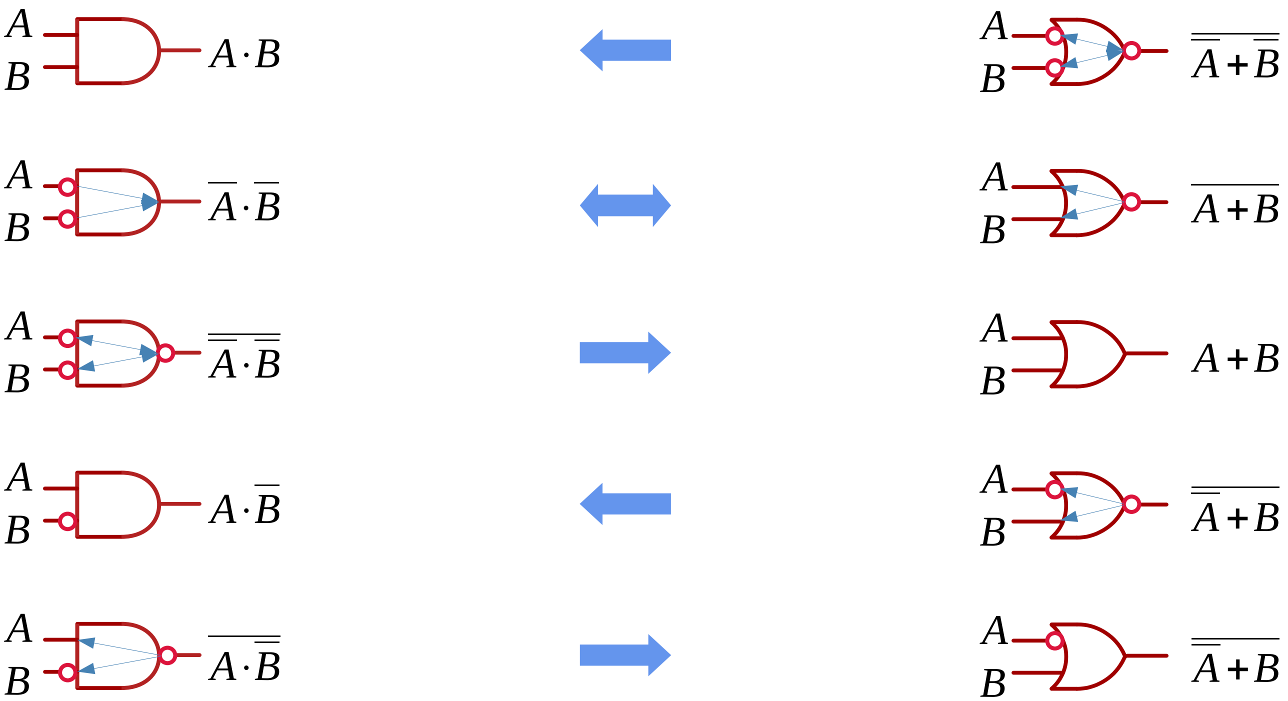


# Bubble Pushing (1)



# Bubble Pushing (2)



# Bubble Pushing Algorithm I

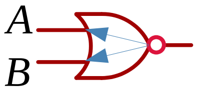
The bubble pushing algorithm is a method

to transform a combinational circuit

that uses NAND, NOR, and NOT gates

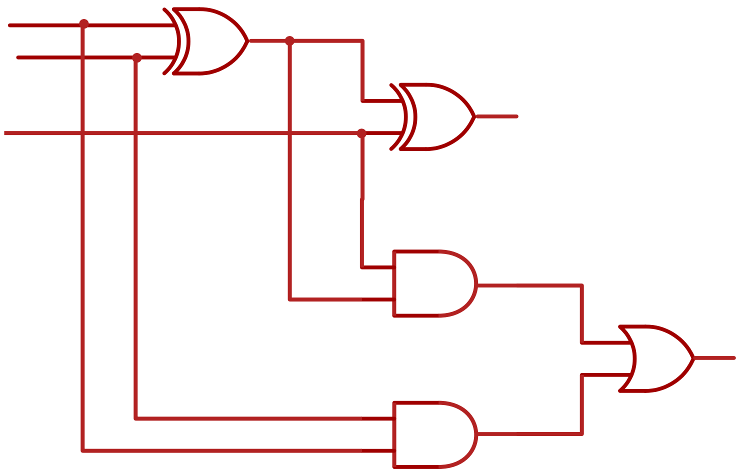
into an equivalent circuit that only uses AND, OR, and NOT gates.

- begin at the output of the circuit and work toward the inputs
- push any bubbles on the final output back toward the inputs
- draw each gate so that bubbles cancel




https://scalable.uni-jena.de/opt/ehp/Bubble_KMaps/assignment.html

# Bubble Pushing Algorithm I

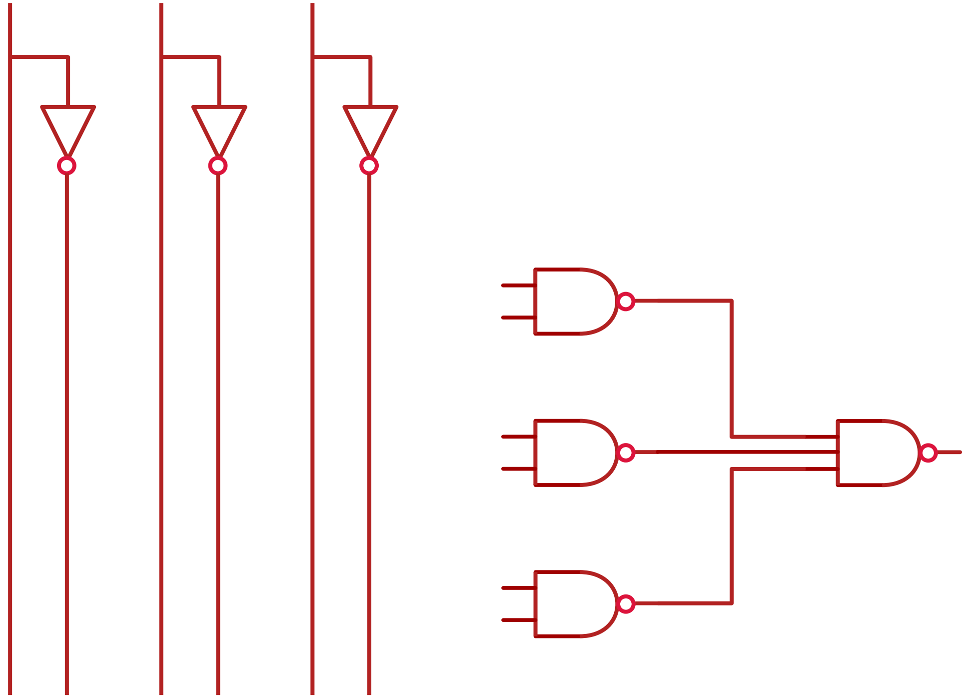


A

https://scalable.uni-jena.de/opt/ehp/Bubble_KMaps/assignment.html

# Bubble Pushing Algorithm II

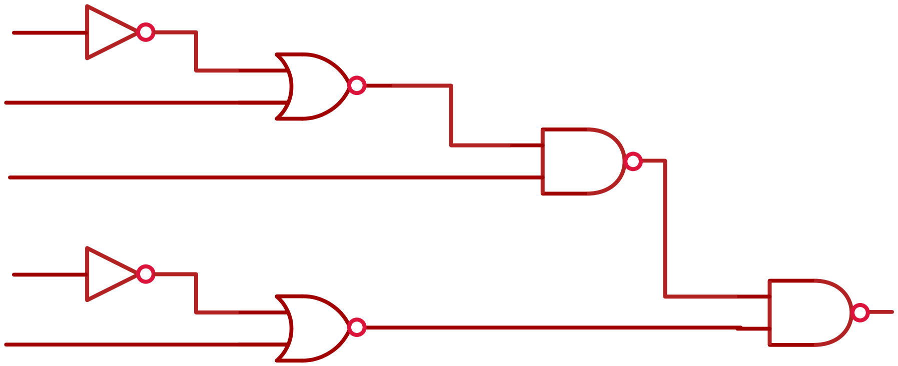
Use the bubble pushing algorithm to derive an equivalent equation in sum-of-product (SOP) form.



https://scalable.uni-jena.de/opt/ehp/Bubble_KMaps/assignment.html

# Bubble Pushing Algorithm III

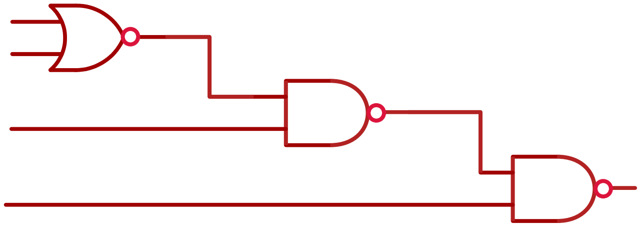
Use the bubble pushing algorithm to derive an equivalent equation
in sum-of-product (SOP) form.

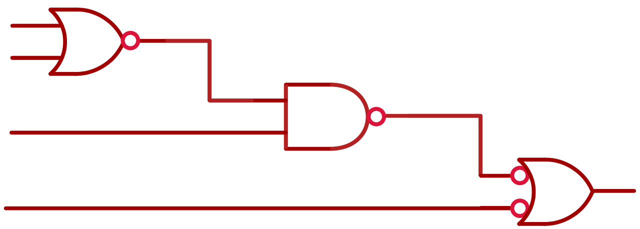


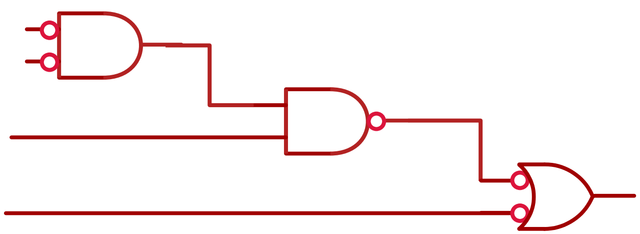
A

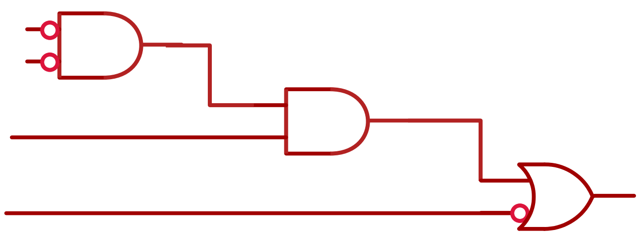
https://scalable.uni-jena.de/opt/ehp/Bubble_KMaps/assignment.html

# De Morgan’s Law (2)

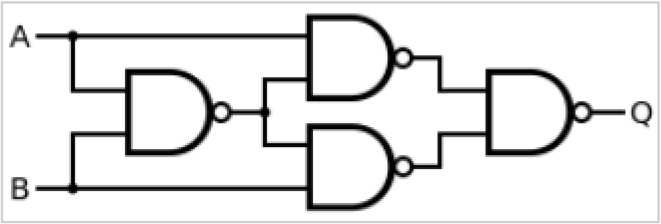




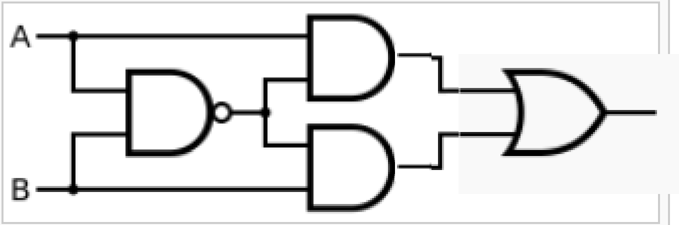




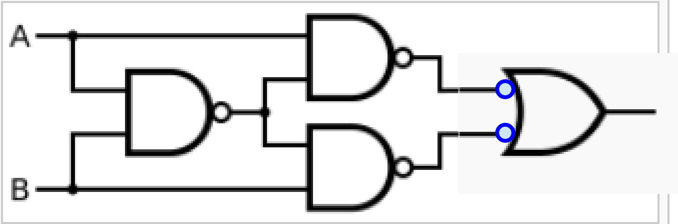
# Bubble Pushing



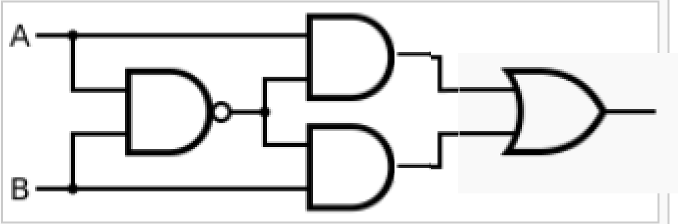
XOR gate circuit constructed using only NAND gates.



XOR gate circuit constructed using only NAND gates.



XOR gate circuit constructed using only NAND gates.



XOR gate circuit constructed using only NAND gates.

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

$$A \cdot \overline{A \cdot B} = A \cdot (\bar{A} + \bar{B}) = A \cdot \bar{B}$$

$$B \cdot \overline{A \cdot B} = B \cdot (\bar{A} + \bar{B}) = \bar{A} \cdot B$$

# References

## References


[1] http://en.wikipedia.org/
[2] M. M. Mano, C. R. Kime, “Logic and Computer Design Fundamentals”, $4^{th}$ ed.
[3] D.M. Harris, S. L. Harris, “Digital Design and Computer Architecture”