BINARY LOGIC AND GATES

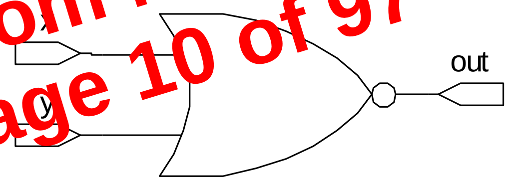
- Binary variables take on one of two values.
- Logical operators operate on binary values and binary variables.
- Basic logical operators are the logic functions AND, OR and NOT.
- Logic gates implement logic functions.

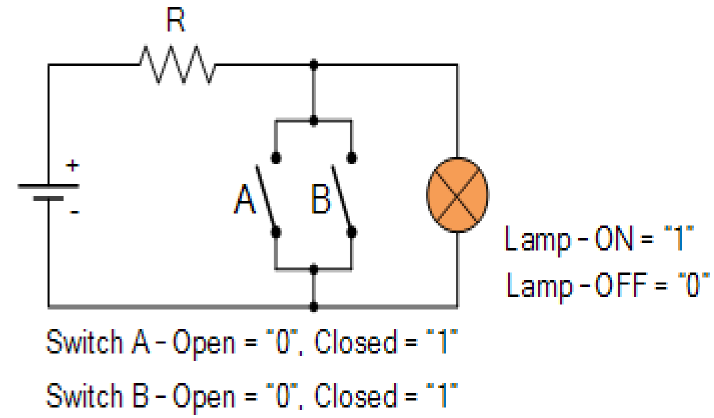
LOGICAL OPERATIONS

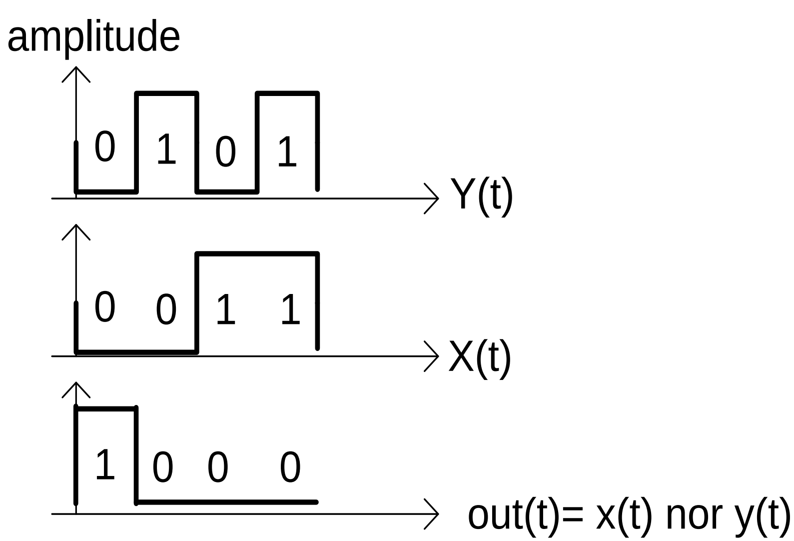
- The three basic logical operations are:
  - AND
  - OR
  - NOT
- AND is denoted by a dot (·).
- OR is denoted by a plus (+).
- NOT is denoted by an overbar (‾), a single quote mark (') after, or (~) before the variable.

NOR

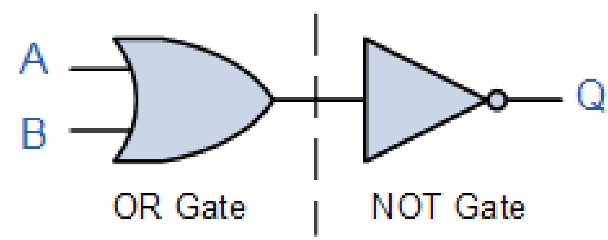
x	y	out = x NOR y
0	0	1
0	1	0
1	0	0
1	1	0





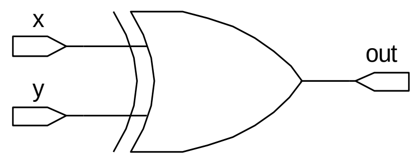


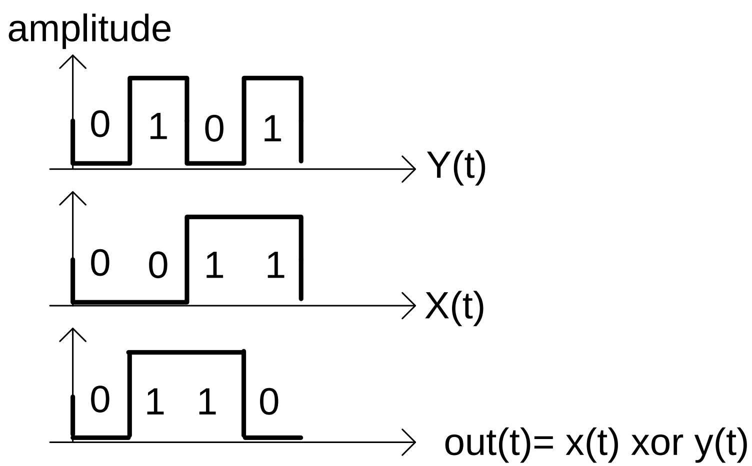
- Output value is the complemented output from an "OR" function.



XOR (EXCLUSIVE OR)

x	y	out = $x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0





- The number of inputs that are 1 matter.
- More than two values can be "xor-ed" together.
- General rule: the output is equal to 1 if an odd number of input values are 1 and 0 if an even number of input values are 1.

TRUTH TABLES

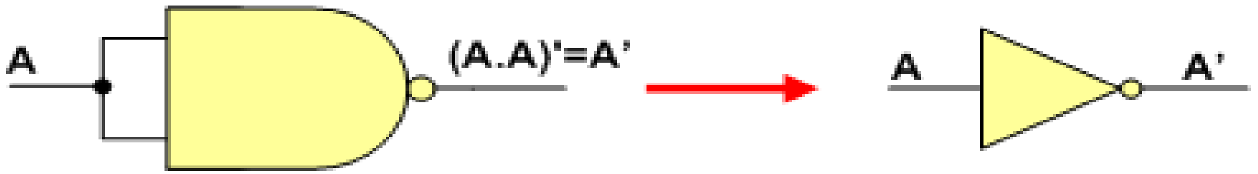
- Used to evaluate any logic function
- Consider $F(X, Y, Z) = X\,Y + \overline{Y}\,Z$

X	Y	Z	$X\,Y$	$\overline{Y}$	$\overline{Y}\,Z$	$F = X\,Y + \overline{Y}\,Z$
0	0	0	0	1	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	1	0	0
1	0	1	0	1	1	1
1	1	0	1	0	0	1
1	1	1	1	0	0	1

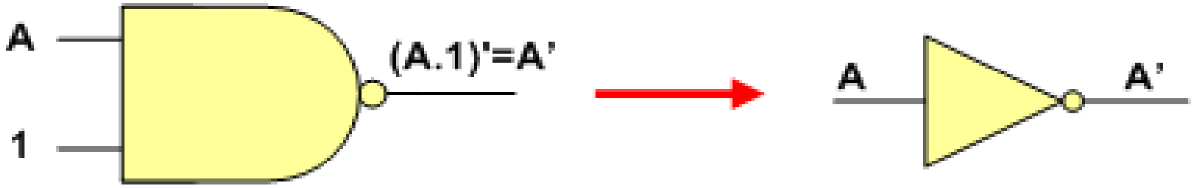
REALIZATION USING NAND

- **NOT GATE**

  All NAND input pins connect to the input signal A gives an output A'.

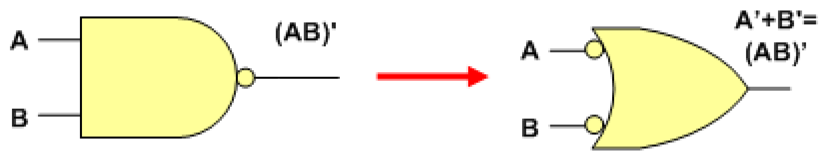


One NAND input pin is connected to the input signal A while all other input pins are connected to logic 1. The output will be A'.

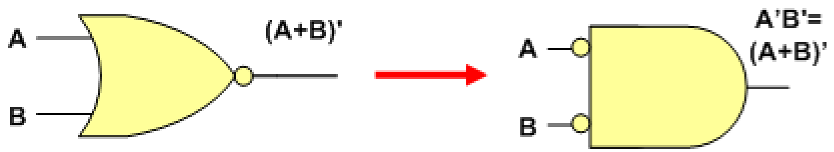


DeMorgan's Theorems

- DeMorgan's theorems provide mathematical verification of:
  - the equivalency of the NAND and negative-OR gates



  - the equivalency of the NOR and negative-AND gates.



DEMORGAN'S THEOREMS

- The complement of two or more ANDed variables is equivalent to the OR of the complements of the individual variables.

NAND Negative-OR

$$\overline{X \bullet Y} = \overline{X} + \overline{Y}$$

- The complement of two or more ORed variables is equivalent to the AND of the complements of the individual variables.

NOR Negative-AND

$$\overline{X + Y} = \overline{X} \bullet \overline{Y}$$

BOOLEAN OPERATOR PRECEDENCE

- **The order of evaluation is:**

1. Parentheses
2. NOT
3. AND
4. OR

- **Consequence: Parentheses appear around OR expressions**
- **Example:** $F = A(B + C)(C + \overline{D})$

Expression Simplification

- An application of Boolean algebra
- Simplify to contain the smallest number of <u>literals</u> (variables that may or may not be complemented)

$$AB + \overline{A}CD + \overline{A}BD + \overline{A}C\overline{D} + ABCD$$

$= AB + ABCD + \overline{A}CD + \overline{A}C\overline{D} + \overline{A}BD$

$= AB + AB(CD) + \overline{A}C(D + \overline{D}) + \overline{A}BD$

$= AB + \overline{A}C + \overline{A}BD = B(A + \overline{A}D) + \overline{A}C$

$= B(A + D) + \overline{A}C$ (has only 5 literals)

- A Simplification Example:

$$F(A,B,C) = \sum(1,4,5,6,7)$$

- Writing the minterm expression:

$$F = \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{C} + A\,\overline{B}\,C + AB\overline{C} + ABC$$

- Simplifying:

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

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

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

$$F = \overline{A}\,\overline{B}\,C + A$$

$$F = \overline{B}\,C + A$$

- Simplified F contains 3 literals

Four-Variable K-Maps

- We can do four-variable expressions too!
  - The minterms in the third and fourth columns, *and* in the third and fourth rows, are switched around.
  - Again, this ensures that adjacent squares have common literals.

			y	y	
	w'x'y'z'	w'x'y'z	w'x'yz	w'x'yz'	
	w'xy'z'	w'xy'z	w'xyz	w'xyz'	x
w	wxy'z'	wxy'z	wxyz	wxyz'	x
w	wx'y'z'	wx'y'z	wx'yz	wx'yz'	
		z	z		

			y	y	
	m_0	m_1	m_3	m_2	
	m_4	m_5	m_7	m_6	x
w	m_{12}	m_{13}	m_{15}	m_{14}	x
w	m_8	m_9	m_{11}	m_{10}	
		z	z		

- Grouping minterms is similar to the three-variable case, but:
  - You can have rectangular groups of 1, 2, 4, 8 or 16 minterms.
  - You can wrap around *all four* sides.

Example: Simplify

$M_0+M_2+M_5+M_8+M_{10}+M_{13}$

- The expression is already a sum of minterms, so here's the K-map:

			Y	
	1	0	0	1
X	0	1	0	0
W, X	0	1	0	0
W	1	0	0	1
		Z	Z	

			Y	
	m_0	m_1	m_3	m_2
X	m_4	m_5	m_7	m_6
W, X	m_{12}	m_{13}	m_{15}	m_{14}
W	m_8	m_9	m_{11}	m_{10}
		Z	Z	

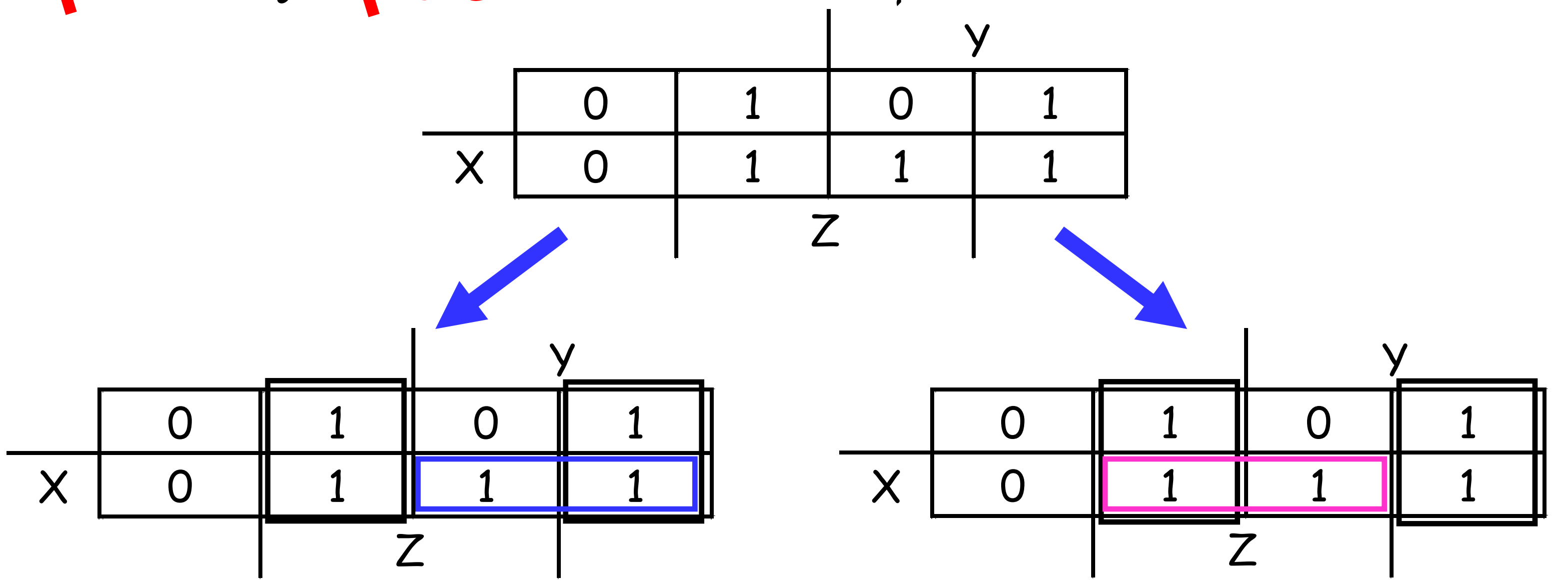
			Y	
	1	0	0	1
X	0	1	0	0
W, X	0	1	0	0
W	1	0	0	1
		Z	Z	

			Y	
	w'x'y'z'	w'x'y'z	w'x'yz	w'x'yz'
X	w'xy'z'	w'xy'z	w'xyz	w'xyz'
W, X	wxy'z'	wxy'z	wxyz	wxyz'
W	wx'y'z'	wx'y'z	wx'yz	wx'yz'
		Z	Z	

- We can make the following groups, resulting in the MSP x'z' + xy'z.

K-Maps can be tricky!

- There may not necessarily be a *unique* MSP. The K-map below yields two valid and equivalent MSPs, because there are two possible ways to include minterm m_7.



y'z + yz' + xy

y'z + yz' + xz

- Remember that overlapping groups is possible, as shown above.

MAXTERM EXAMPLE

		(A+B)	(A+B`)	(A`+B`)	(A`+B)
		A`B`	A`B	AB	AB`
C	C`		0	0	0
C`	C	0		0	

$f(A,B,C) = \prod M(1,2,4,6,7)$

=(A+B+C`)(A+B`+C)(A`+B+C) )(A`+B`+C) (A`+B`+C`)

Note that the complements are (0,3,5) which are the minterms of the previous example

Four Variable Example
(a) Minterm form. (b) Maxterm form.

$$f(a,b,Q,G) = \sum m(0,3,5,7,10,11,12,13,14,15) = \prod M(1,2,4,6,8,9)$$

QG \ ab	00	01	11	10
00	0: 1	4:	12: 1	8:
01	1:	5: 1	13: 1	9:
11	3: 1	7: 1	15: 1	11: 1
10	2:	6:	14: 1	10: 1

(a)

QG \ ab	00	01	11	10
00	0:	4: 0	12:	8: 0
01	1: 0	5:	13:	9: 0
11	3:	7:	15:	11:
10	2: 0	6: 0	14:	10:

(b)